Remote Computing

With pcAnywhere you can connect one PC to another using either of the following methods:

- A remote control session, where a pcAnywhere remote PC connects to and controls a pcAnywhere host PC. In addition to accessing files and applications on the host, you can also transfer files during a remote control session.
- A remote networking session, where a PC dials into a network, and becomes a remote workstation on the network.

Introducing pcAnywhere

pcAnywhere offers a total communications solution for your remote control, file transfer, and remote networking needs.

With pcAnywhere you can:

- Remotely control one PC from another, as long as both PCs are running pcAnywhere software.
- Automatically transfer files and synchronize folders between two PCs.
- Connect to a variety of online services.
- Become a workstation on a network using a remote networking connection.

You can make connections using telephone lines, networks, or a combination of the two. You can also make direct connections between two PCs using either a serial or parallel cable. For example, you may want to transfer files between your laptop and desktop PC.

Key features of pcAnywhere 9.0

Wizards

pcAnywhere wizards are a series of menus that assist in configuring connection items by prompting you for the required information.

SpeedSend™ file transfer

SpeedSend™ improves the performance of file transfer sessions by comparing the two files and transferring only the new data.

Optimized Desktop Control

Desktop optimization improves the performance of a remote control session by disabling any desktop wallpaper, pattern, or screen saver configured on the host PC.

Dial-Up Networking

A dial-up network connection allows any PC to become a remote workstation on the network.

TAPI Modem Support

pcAnywhere automatically detects the modem selected in the Windows 9x dialog box and uses that modem for pcAnywhere modem connections.

Parallel Port Connections

Direct connections between two PCs, using the parallel port enhances the performance of the connection.

Drag and Drop Capability

You can drag pcAnywhere connection items to the desktop. Double-click the item to automatically run the program.

Multiple Host Configurations

A pcAnywhere host can be set to use different settings. For example, you can have a host that uses a modem as the connection device, and another that uses a network connection device. It is not necessary to edit the host's configuration each time you want to change the connection device. Simply select the host with the configuration you want to use. Other settings, such as locking the PC when waiting for a call or blanking the host screen after connecting, can also be set independently for each host item.

Host Callback

Use pcAnywhere's callback feature to reverse phone charges or increase security on the host PC. Callback can be set for all callers or individual callers. When the remote caller connects to the host, the host identifies the caller in the caller list, cancels the connection, then calls back the remote caller using the number stored in the caller's list.

AutoTransfer

AutoTransfer allows you to create a file containing file transfer commands to automatically send, receive, or synchronize files on the host PC.

Encryption

Encryption increases security using Microsoft's cryptography API.

NT Caller Security

NT Caller Security feature integrates pcAnywhere remote callers with Windows NT User and Group Security.

Voice First Connections

Voice First allows switching between voice and data during a single phone call.

Host Conferencing

Host conferencing allows multiple remotes to connect to and view the activities on one host.

What's New in version 9.0

Central Management Systems Integration

Integration with central management systems, such as Norton System Center, Tivoli-TME, and CA-TNG, provides utilities for remote installations, remote starting and canceling of hosts, and central management of pcAnywhere connection items.

Centralized Logging

Central logging allows for exporting pcAnywhere events into standard database formats and also supports simple network management protocol (SNMP).

Improved File Transfer Interface

Easier file management and file transfer control with a new explorer-like interface.

Administrator Customized Installations

Custom configuration allows an administrator to pre-configure many pcAnywhere installation and application settings before distributing to the workstation environment.

Directory Services Support

Register pcAnywhere 9.0 hosts with an LDAP, or NDS directory service to improve remote location of hosts.

Enhanced Security

- Warning prompt when making an unsecured connection.
- Windows NT Authentication
- Windows 95 Domain Authentication
- Checkpoint VPN (Virtual Private Networks) remote client software provided on pcAnywhere CD.

Expanded OLE Automation

pcAnywhere 9.0 host and remote connection items are now available from within other applications.

System Requirements

This version of pcAnywhere is designed for any of the following 32-bit operating systems:

- Windows 95
- Windows 98
- Windows NT
- Windows Internet Explorer 4.01 (for central management features)

Hardware Requirements:

- 486DX / 66MHz or greater
- 16MB RAM
- 50MB free disk space for optimal performance
- VGA video minimum
- TAPI, ISDN, or Memory modems.

What can I do in a remote control session

After establishing a remote control connection with a host PC, the remote user can operate the host PC as if they were physically at the host PC.

The remote user can:

- Use programs and files on the host PC
- Troubleshoot a technical problem on the host PC
- Transfer files quickly between the two PCs.
- Access programs and files on a network.

How do PCs connect to each other

You can connect using pcAnywhere through the following connection devices:

- A modem, if the other PC also has one.
- A direct connection using a null modem cable attached to either the serial or parallel ports of both PCs.
- A network, if both PCs are attached to it.

Parallel port connections are not supported in Windows NT.

What is dial-up networking?

Dial-Up networking (Windows 9x), or Remote networking (Windows NT), is a modem connection to an organization's network. Once the connection is made, the remote PC can log on to the network and act as if it were cable-connected to the network, accessing any file or application that the remote user has permission to use.

What is remote control?

A remote control connection allows a remote PC to connect to a host PC and use the host PC as though the user were at the host site, accessing any file or application on the host PC or the network. Remote control minimizes information transfer over the telephone line because the data transfers are limited to keyboard, mouse and screen information.

What is the difference between dial-up networking and remote control?

While both technologies make connections to a network, using *remote control technology*, applications are executed on the host PC, making it unnecessary to transfer executable files and other large files to the remote PC. The only data being communicated to the remote PC is the screen display, keyboard input and mouse input of the host PC.

When you connect to a network using *dial-up networking technology,* applications are executed on the remote PC. All information and data required by the application must be communicated over the modem connection to the remote PC. Even with high-speed modems, connections are slower than direct network connections, and performance could be an issue. Dial-up networking connections are better suited to applications that perform the majority of processing on the remote PC, making limited queries to the network server.

Administrator Installation

The pcAnywhere administrator can customize the network version of pcAnywhere and control the availability and use of connection items.

The pcAnywhere administrator can:

- Allow users to name their computers.
- Define a default folder for connection items.
- Allow a user to select a different folder.
- Protect connection items with a password to prevent users from modifying connection item properties.

pcAnywhere connection items

Connection items appear in the main window for each of the buttons on the action bar. A connection item represents a file containing connection device information and security settings used for a connection.

Every connection item has a properties sheet associated with it containing the settings configured for the item.

Connection items can be protected from unauthorized use by assigning a password to the item. You can use the same password for all connection items, regardless of what mode they are in. Using the same password is easier than having to remember multiple passwords. A password can protect the item from being modified, viewed, or executed.

All pcAnywhere connection items can be dragged or copied from the main window to the Windows desktop or a folder.

To create connection items

1 Click an action button on the pcAnywhere action bar.

2 Double-click Add action Item to run the connection Wizard.

3 Follow the on screen instructions to complete the wizard.

The wizard creates a connection item using pcAnywhere's default settings.To view or modify the default settings, right-click the connection item and choose Properties from the context menu.

 Click the Help button on any property page for detailed information on the available settings.

action

Be A Host

Remote Control

File Transfer

Be A Gateway

Logging pcAnywhere Sessions

pcAnywhere logging options are available from the Tools menu. You can log various host, remote, and file transfer activities to any one of the following:

- **Windows NT Event Log** - reports application events to the NT Event Log. The events can then be viewed and used for diagnostic and monitoring purposes. You can monitor all pcAnywhere host, remote, and file transfer activities.
- **SNMP** (Simple Network Management Protocol) Options- a Windows component allowing a computer to be monitored remotely with third party management tools.
- **pcAnywhere log file** - allows you to capture information about remote control sessions for historical or security purposes. Information in the log file is viewed by generating an activity log report.
- **Host Session recording file** - allows you to create a video recording of all activities occurring during a session with a host PC. This file can be played back after the session ends and used to troubleshoot a complex program or procedure run on the host PC.

To select a logging option:

- Choose Logging Options from the Tools menu.

Click the Help button on the logging property pages for detailed information on each logging option.

Application Options Overview

Application options are global settings used in pcAnywhere's various modes of operation. pcAnywhere uses default settings for these options unless the user customizes them. These default settings are the most commonly used and it may not be necessary to change them.

Options configured in the Application Options properties sheet affect all sessions on the PC. Some options can be temporarily changed during a remote control connection. These online changes affect only the current session and do not modify the settings configured in the Application Options properties.

 Click Help on the application options property pages for more information on these options.

Running scripts

pcAnywhere includes a script language that allows you to:

- Perform operations automatically when calling online services.
- Automate some procedures during remote control services.

Use scripts to automate:

- Running programs.
- Transferring files.
- Performing arithmetic operations.

Converting data

To use data conversion:

1 Choose Data Conversion from the Tools menu.

2 Select the data conversion feature you want to run.

3 Follow the Data Conversion Wizard's instructions.

Use data conversion to:

- Convert configuration files from earlier versions of pcAnywhere so that they can be used as connection items in pcAnywhere 9.0.
- Export configuration files from a earlier version of pcAnywhere to a text data file for report generation.
- Import data from a pcAnywhere text data file into pcAnywhere 9.0 connection items.

Note: If you choose to convert pcAnywhere files, you are prompted to enter the folder containing source files, the version of pcAnywhere in which they were created, and, if applicable, the User ID identifying the files. If you used User Ids in earlier versions of pcAnywhere, you can convert only one user's file at a time.

Remote Printing

When a remote PC is connected to a host PC, all activities occur on the host PC, including printing.

In order to direct a print command to the remote PC's printer during a session, you must first install the remote PC's printer drivers on the host PC.

To print files at the remote PC during a session:

On the host PC:

1 Choose Application Options from the Tools menu.

2 Click the Remote Printing tab.

3 Click Add Printer to add a remote printer definition to the list.

4 Do one:

- Select the remote printer driver from the list of Currently Installed Drivers and click next.
- Select the manufacturer and model of the remote printer from the Manufacturer's and Printers list boxes and click Next.

5 Type a name that identifies the owner or location of the remote printer and click Finish. The remote caller selects this name to print to the local printer during a remote control session.

Note: If the remote printer driver is not listed in the Manufacturers list box, install it following the printer manufacturers instructions. After installing the driver, it appears on the Currently Installed Driver list.

DOS Sessions

When the host is running DOS in full screen mode, or when the host user has exited to DOS, the remote PC may experience problems reading and displaying the host screen.

To change settings for DOS sessions:

1 Choose Application Options from the Tools menu.

2 Click the DOS Sessions tab.

3 Configure the DOS session options you want to use.

 For detailed information on DOS Session options, click the Help button on the DOS Sessions dialog box.

Remote Control connection item

Double-click to call a host PC.

Right-click to change this item, to rename it, or to initiate a call.

Host connection item

Double-click to wait for a call from a remote PC.

Right-click to change this item, to rename it, or to initiate a connection by calling a remote.

Online Service connection item

Double-click to connect to an online service.

Right-click to change this item, to rename it, or to delete it.

The pcAnywhere action bar

pcAnywhere Action Bar

pcAnywhere Caller Item

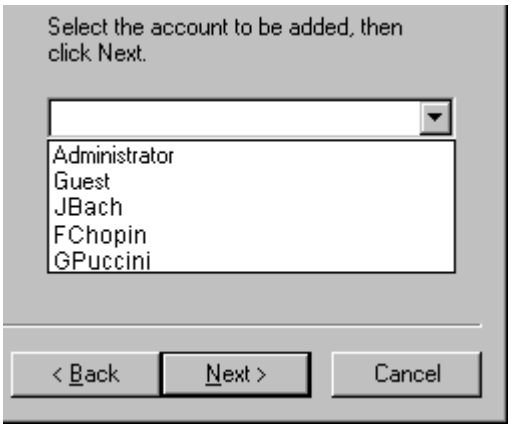
Caller items contain connection information, including passwords, for remote users accessing the host PC. They are created and stored on the host PC.

Right-click to modify options, rename, or delete.

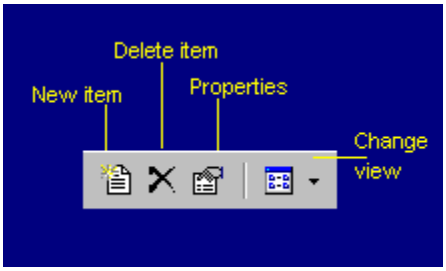
Windows NT User List

The users or groups listed are maintained by the Windows NT administrator.

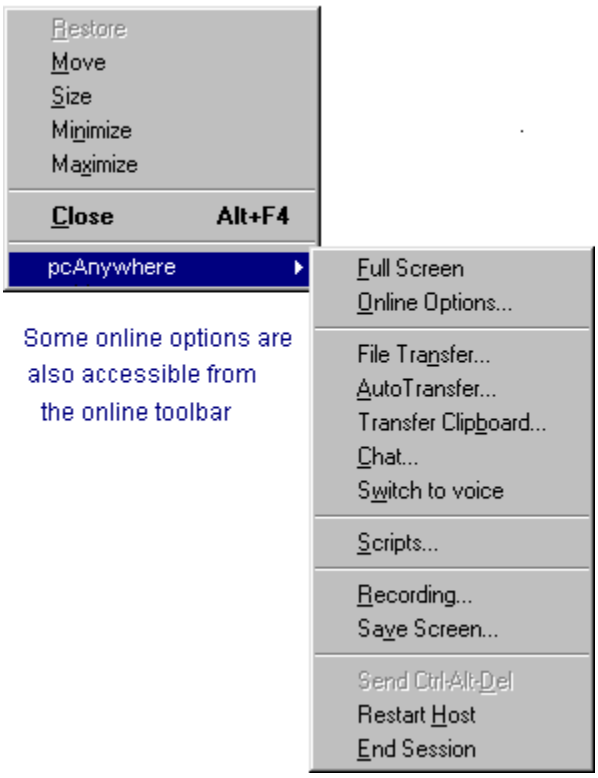
You must have NT Administrator privileges to add or delete users or groups.



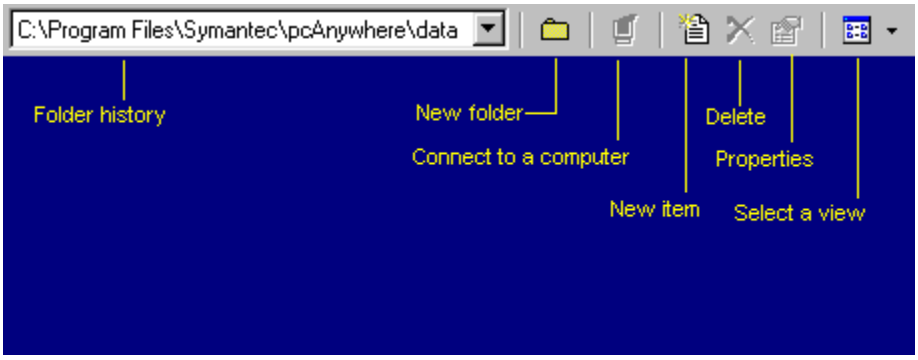
Caller Toolbar



Remote online menu



pcAnywhere toolbar



To record a session

You can start recording a session in two ways:

- Automatically upon connection: When recording is enabled in the Automated Tasks property page of the remote control connection item, recording begins immediately after the connection is established.
- During a session: A recording option on the online menu allows you to start and stop recording any time during the session..

To start recording automatically:

1. Right-click a remote control connection item and choose Properties from the drop-down menu.
2. Click the Automated Tasks tab.
3. Check Record Session in File For Later Playback.
4. Type a name for the recorded file. If you leave this text box blank, you are prompted for a filename when you stop the recording or end the session.
5. Click OK.

Recording starts immediately after making a connection with this connection item.

To start recording during a session:

1. Choose Recording from the remote control online menu.
2. Do one:
   - Choose an existing record file to add the current recorded session to the end of that file.
   - Type a new filename to save the session in a new record file.
3. Click Save to save the session to the selected recording file.

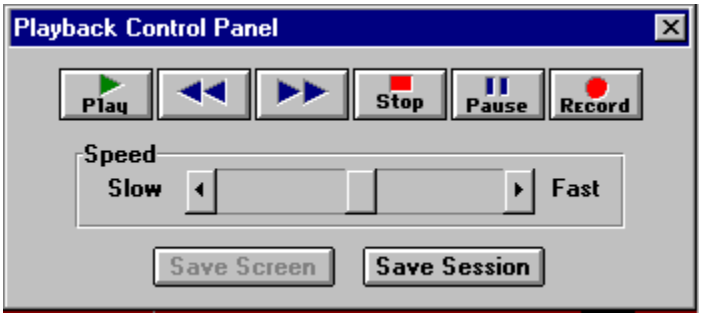
To stop recording:

- Choose Recording from the remote control online menu.

To playback a recorded session

1. Choose Playback Sessions/Screens from the Tools menu.
2. Select the recording file (*.RCD) you want to play back.
3. Choose the playback options you want to use and click OK.
4. Use the Playback Control Panel to control the session playback.

Playback control panel



Click **Save Screen** during playback to save the current screen.

Click **Save Session** to save the recorded session in a raw binary file.

To choose a connection device

1 Right-click the connection item for which you want to choose a connection device.

2 Choose Properties from the drop-down menu.

3 In the Connection Info tab, check one:

- A TAPI modem definition that Windows placed on the pcAnywhere device list.
- A COM port for a modem or serial cable connection.
- A network protocol for a network connection.
- A parallel port for a parallel cable connection.

4 [optional] To customize the device, click Details and choose any options.

Note: Both PCs must be running Windows to make parallel port connections. To communicate in either direction, use a bi-directional parallel cable and configure the parallel port in the BIOS for bi-directional communication.

Modem Tip

Modem Tip

Both Windows 95 and Windows NT use Microsoft's Telephony Application Programming Interface (TAPI) to detect and configure modems. If your operating system supports TAPI devices and has added your modem to the top of the device list on the Connection Info tab, you should choose it for modem connections. It is not necessary to customize the device.

Please refer to your operating system's documentation for information on how to detect and configure TAPI modem devices.

To customize a connection device

1 Right-click the connection item for which you want to choose a connection device.

2 Choose Properties from the drop-down menu.

3 In the Connection Info tab, check the connection device you want to customize.

4 Click Details.

5 Select the options to customize the device.

- For detailed information on the cutomized settings, click the Help button on the Details dialog box.

To customize an ISDN connection device

1 Check ISDN via CAPI 2.0 in the Connection Info property page of the connection item.

2 Click Details.

3 Check Attempt Channel Bonding if you want pcAnywhere to use two B channels for the connection. The connection is made using one B channel if the second channel is not available at the time of the connection.

4 In the Seconds to Wait After Dial box, type the maximum time pcAnywhere allows for a successful connection after a number has been dialed.

To set up connection devices for a gateway

On the gateway PC:

1 Click Be a Gateway on the pcAnywhere action bar.

2 Right-click the gateway connection item whose connection devices you want to set up.

3 Choose Properties from the drop-down menu.

4 Check the incoming device for this gateway.

5 Check the outgoing device for this gateway.

6 [optional] To customize each device, click the name of the device and click Settings.

incoming device

The device used by the gateway PC when receiving calls. The remote PC calls this device to connect to the gateway.

outgoing device

The device the gateway uses to access a waiting host PC.

To customize a network to use a gateway

1 Click either Remote Control or File Transfer on the pcAnywhere Action Bar.

2 Right-click the connection item that you want to configure.

3 Choose Properties from the drop-down menu.

4 In the Connection Info tab, check either IPX, SPX, NetBIOS, Banyan VINES, or TCP/IP.

5 Click Details.

6 Check Use Gateway in the Details dialog box.

7 [optional] Type the name of the pcAnywhere gateway you want to use.

8 [optional] Type the name of the gateway's class.

To customize a COM port for parallel connections

1 Right-click the connection item for which you want to choose a connection device.

2 Choose Properties from the drop-down menu.

3 In the Connection Info tab, check the port you want to customize (COM1, COM2, COM3, or COM4).

4 Click Details.

5 Select the connection preferences options to customize the device.

For detailed information on setting the connections preferences for your COM port, click the Help button on the Details dialog box. .

To name your computer in pcAnywhere

1 Choose Application Options from the Tools menu.

2 Select User defined:

3 Type a name that you want pcAnywhere to use to identify your PC in a remote control session.

Note: The User Defined option may not be available to you if your are using a network version of pcAnywhere.

To select Color Scale

On the remote PC:

1 Choose Application Options from the Tools menu.

2 Click the Remote Operation tab.

3 Select the ColorScale option you want to use to display colors on the host PC during a remote control session.

To use this option effectively, the color palette on the host PC must be set to 256 colors or greater.


About ColorScale

About ColorScale

Selecting fewer colors in this option results in improved performance.

To select a ColorScale option during a remote control session, click the online options icon in the online toolbar.

To select ColorScale during a session

On the remote PC:

1 Click the online options icon in the online toolbar.

2 Choose the ColorScale option you want to use.

3 Click OK.

The host window refreshes using the selected ColorScale option.

To use this option effectively, the color palette on the host PC must be set to 256 colors or greater.

To reduce host desktop area

On the remote PC:

1 Choose Application Options from the Tools menu.

2 Click the Remote Operation tab.

3 Check **Reduce Host Desktop to Match Remote**.

 This option improves the performance during a remote control session by causing the host to reduce it's resolution to match the resolution used on the remote PC.

This option improves the performance of a session by reducing the resolution of the host PC to the same resolution used on the remote PC.

This option can also be enabled during a session from the Online Options menu.

This option improves the video performance during a session by disabling Microsoft's Active Desktop feature on the host PC.

To use screen scaling on host PC

On the remote PC:

1 Choose Application Options from the Tools menu.

2 Click the Remote Operation tab.

3 Check the Screen Scaling option.

- Screen scaling causes the host screen to scale down to fit in the remote's window. If this option is not selected, the remote user may need to scroll to view portions of the host screen.

To add a TCP/IP host

1 Choose Application Options from the Tools menu.

2 Click the TCP/IP tab.

3 Type any of the following identifiers for the TCP/IP hosts you want to connect to:

4 **pcAnywhere host name** - Use if the host is located on the local subnet. The host name appears in the Network Hosts dialog box.

5 **DNS name** - Use if the network administrator has assigned DNS names and IP addresses to the host PCs.

6 **IP address** - Use to connect to a host regardless of the subnet it is located on. The host IP address appears in the Network Hosts dialog box.

7 **Specific group IP address** - Substitute 255 for the last portion of an IP address to display all hosts having an IP address containing the first three portions of the group address. For example, an address of 120.45.62.255 causes pcAnywhere to list all hosts with an IP address beginning with 120.45.62.

- Use the group IP address to connect to a host that is connected to the Internet through an Internet service provider.

8 Click Add.

To use directory services

Note: Before you can configure pcAnywhere to use directory services, the directory server entries must be configured by a network administrator.

To configure pcAnywhere for directory services:

1 Select Network Options from the Tools menu.

2 Select the Directory Services tab.

3 Click the Add button.

4 Type a Display Name that clearly describes this Directory Server.

5 Type the address of the Directory Server in the Directory Server text box.

6 Type the common name and password to use when authenticating to the directory server.

7 Click Advanced to configure the port number and the search base of the directory tree.

8 Click OK.

pcAnywhere attempts to connect to the directory server and search for the entry that was specified in the Name field. If multiple entries are found, the user can choose from a list of entries. After the entry is identified, pcAnywhere stores the Distinguished Name in the registry for easy identification by the host, and labels the entry as 'verified'.

If verification fails, check the network connection or the TCP/IP configuration.

To configure the host for directory services

1 Right-click on a host connection item and select Properties from the drop-down menu.

2 Click the Settings tab.

3 Check Use Directory Services.

4 Select the Directory Server from the list box.

The pcAnywhere host searches the directory server for the Common Name. If it finds a corresponding entry it updates it with the connection information and current status of the host. If the host status changes, the directory server is updated to display the current status of the host.

To configure the remote for directory services

1 Right-click on a remote connection item and select Properties from the drop-down menu.

2 Click the Settings tab.

3 Click Use Directory Services.

4 Select the Directory Server from the list box.

5 Set initial filter settings.

The filter settings apply to the current Directory Server and are stored with the server's other configurations. If you select a different Directory Server, it will contain different filter settings. These settings can be modified from the directory service's browser window.

Dial-up Networking

Dial-up networking allows a PC to connect to a network by dialing into a dial-up network server. Using remote networking, you can call from your home PC, or laptop from any remote location, connect to the office network, and log on. The remote PC works as if it were cable-connected to the network, giving the remote user access to any file or application that the remote user has access privileges to use.

To configure Dial-up network settings

On the remote PC

1 Right-click a remote connection item and choose Properties from the drop-down menu.

2 Select TCP/IP from the device list and click Details.

3 Choose one:

▪ Connect to a pcAnywhere Host Using a Local Area Network—Select this option if you are connecting to a host using a network protocol.
Connect to a pcAnywhere Host Using Dial-up-Networking— Select this option to specify modem dialing properties to use when calling the host PC.
For modem connections:
Do one of the following:

1 Choose a dial-up configuration from the drop-down list box.

2 Click Add to add a new dial-up configuration.

3 [optional] Click Properties to add or modify your location or dialing properties.

4 Set the number of redials pcAnywhere attempts to make a connection. Type 0 to disable the redial feature.

5 Set the number of seconds pcAnywhere pauses between redial attempts.

6 Type the name and password you use to log on to the server

7 [optional] Type the domain you are connecting to.

To customize a TCP/IP device

On the remote PC:

1 Do one of the following:

- Double-click the Add Remote Control Item to add a new remote control connection item and right-click on the new item.
- Right-click an existing network connection item.

2 Choose Properties from the drop-down menu.

3 In the Connection Info tab, check TCP/IP for the connection device.

4 Click Details.

5 Choose the Dial-Up networking options to use for this connection item:

- Connect to a pcAnywhere host using a local area network—select this option if you are connecting to a host using a network protocol.
- Connect to a pcAnywhere Host Using Dial-Up Networking—select this option to specify modem dialing properties to dial in to the host machine over an Internet connection.

Modem connections:

1 Do any of the following procedures:

- Choose a dial-up networking configuration from the drop-down list box.
- Click Properties to change the settings of a dial-up network configuration.
- Click Add to add a new dial-up network configuration.

2 Choose the number of time this connection item attempts to make a connecction.

3 Choose the number of seconds to wait between redial attempts.

4 Type the name and password used to connect to the network or Internet Service Provider.

5 [optional] Type the Domain you are connecting to.

6 Select the number of minutes to allow for inactivity before disconnecting the connection.

- In Windows NT, you can include the domain name in addition to the login name to automatically locate the user on a specific domain.

To use full screen display

On the remote PC:

1 Choose Application Options from the Tools menu.

2 Click the Remote Operation tab.

3 Check Local Full Screen Display and click OK.

- Local full screen display displays the host image full-screen on the remote PC rather than a window. You can press Alt+Enter to return to a window display at any time during the session.

To set Host Active window tracking

On the remote PC:

1 Choose Application Options from the Tools menu.

2 Click the Remote Operation tab.

3 Check Host Active Window Tracking and click OK.

- Host active window tracking automatically moves the remote's view to any currently active window on the host PC during a session.

Activity Log Processing

When you choose pcAnywhere logging from the Logging Options property page, the data captured in the log file can be formatted into a report.

To create a pcAnywhere log report:

1 Choose Activity Log Processing from the Tools menu.

2 Click Report.

3 Select the log file from the Choose Input Log File list box and click Open.

4 Choose one of the following formats for the report:

- **Fully formatted**: formats log information into a report format, with a summary of events.
- **Data Only - Comma Delimited**: formats information for use in database applications or other programs requiring delimiters between fields. Each item is enclosed in double quotes, followed by a comma. Month, date, and year are separated by slashes. Each entry is following by a carriage return.
- **Data Only - Fixed Fields**: formats information in columns of fixed width.

5 Type a name for the report and click OK.

To archive pcAnywhere log data:

1 Choose Activity Log Processing from the Tools menu.

2 Click Archive/Delete.

3 Select the log file from the Choose Input Log File list box and click Open.

4 Enter the date range of the data you want to archive.

5 Check Delete Log Entries from source to remove the archived entries from the log file.

- Removing the archived information from the log file reduces the size of the log file.

6 Check Copy Log Entries to archive file.

7 Type a name for the archive file and click Save.

To configure Dial-up networking

To configure a dial-up network connection item:

1 Click Remote Control on the pcAnywhere Action Bar.

2 Do one of the following procedures:

- Right-click a remote control connection item, choose Properties from the menu, and select TCP/IP as the connection device.
- Create a new remote control connection item and choose TCP/IP as the connection device.

3 Click Details and enter the following information.

- **Connect to a pcAnywhere host using a local area network:** Select this option to make network connections to a server or ISP (Internet Service Provider).
- **Connect to a pcAnywhere host using a modem:** Select this option to make a modem connection to a server or ISP (Internet Service Provider).
- **Use the following Dial-Up network connection**: Choose the dial-up networking configuration to use to make a connection to a network server or Internet Service Provider (ISP).

  [optional] Click Add to configure a new dial-up network configuration.

  [optional] Click Properties to view the dialing properties of the selected configuration.
- **Number of times to attempt connection**: Set the number of time the remote redials to try and connect to the host
- **Number of seconds to wait between attempts**: Set the number of seconds to wait between redial attempts.
- **Log on information**: Enter the User name, password, and domain [optional], used to connect to the host.
- **Disconnect if idle for**: Set the number of minutes to allow for inactivity before disconnecting the session.

General buttons -- probably not needed...

Closes property sheets and saves any changes you have made.

Closes property sheets without saving any changes you have made.

(Shared) Connection Info page

Check to choose a connection device for pcAnywhere sessions.

Click to set options for the device you have selected.

Click one to choose an outgoing connection device for this gateway PC.

Click to set options for the device whose name you have selected.

(Shared) Protect Item tab

Type password to protect this connection item.

Re-type password to protect this connection item.

Makes password above necessary to view the properties of this connection item.

Makes password above necessary to make a connection using this connection item.

Makes password above necessary to change any of this connection item's properties.

Starts your Windows screen saver when a host is waiting for a call. Assign a password to the screen saver to lock the host when waiting for a call. You enable screen savers in the Windows 95 Display Properties and in the Desktop dialog box in the Windows NT Control Panel.

Prevents unauthorized users from canceling the waiting host by locking the keyboard on the NT Workstation

Remote Control item - Settings tab

Displays name of this remote control connection item.

Identifies host computer for network connections. (To enable this option, click the Connection Info tab and select a network connection device.)

Check this option to search for a pcAnywhere host on the selected directory server. The directory service must be configured before you can select it. Choose Network Options from the Tools menu to configure a directory service.

You can refine the search for a pcAnywhere host by using any of the following:

- **Name:** To search for a specific user name.
- **Mail:** To search for a specific email address.
- **Organization**: To search for all users in a specific organization.
- **Department:** To search for all users in a specific department.

- You can use wild cards in any field to search for a group of users. For example: type *jones* to list all users having 'jones' in the name, or type *xyz*.* to search for any organization containing xyz.

Telephone number of host's modem, needed for modem connections only. (To enable this option, click the Connection Info tab and select a modem connection device.)

The phone number of the modem host PC. If a host phone number is provided on the Settings page, you are prompted for it when you run the remote control connection item.

Country code prefix for the location you are calling.

Area code for the location you are dialing.

Use this field when you do not want to use the dialing properties configured for your present location. Enter the phone number of the host, including area code and any required dialing prefix.

Check to enter your assigned login name and password. pcAnywhere then automatically sends this information to the host upon connection.

Type assigned login name for access to host.

 If you are calling a Windows NT host, you can include the domain name in addition to the login name. For example, type *domain**username* to automatically locate the user on a specific domain.

Type assigned login name for access to host.

 If you are dialing into a Windows NT host, you must include the domain name in addition to the login name.

Type assigned password for access to host.

Type assigned password for access to host.

Re-type assigned password for access to host. (Same as Password above.)

Check to run a software program on the host PC automatically on connection.

Dials the area code or country code, and phone number using the dialing properties configured for your current location.

Remote control connection item: Automated Tasks tab

Enables you to record remote session for later playback in file you select. (If you check this option and do not enter a filename in the area below, you will be prompted for one when making a connection.)

Type name of file in which to record of remote session for later playback. (If you do not enter a filename here, you will be prompted for one when making a connection if you have checked the option above.)

Click to view and select existing files containing records of remote sessions.

Runs selected script or AutoTransfer procedure after connection.

Click to run a script file after connecting.

Click to run an AutoTransfer procedure after connecting.

Name of script file to run.

Click to view and select existing AutoTransfer procedure.

Click to view and select existing script file.

Enables you to save session statistics in a log file for later reports of pcAnywhere activity.

Caller item, Settings tab:

Displays name of caller item.

Type to assign login name of remote user. (This is required.)

Type to assign password of remote user.

Re-type password of remote user. (Same as Password above.)

Caller item, Callback tab:

If checked, host disconnects at connection time and calls back remote user.

Type number to dial when calling back remote user.

Caller item, Advanced tab:

Click to give the caller unrestricted access to the host.

Click to select specific access rights for this caller.

Enables remote user to blank screen of host PC during session.

Enables remote user to determine at termination of session whether host PC remains accessible or not.

Enables remote user to restart host PC during session.

Enables remote user to send files to the host.

Enables remote user to get files from the host.

Enables remote user to terminate processes with the keystrokes CTRL+Break.

If checked, limits time remote user can maintain connection to the host.

Sets time limit of session, if you have checked Limit time allowed per session.

Type a command to be executed upon connection.

If checked, causes termination of session if remote user is inactive for time exceeding that set in host connection item's Security Options tab.

Saves statistics of sessions with remote user in file on host PC.

Opens dialog box with Drive Access options to set access rights for remote user.

Caller Drive Access dialog box (accessed from Caller item's Advanced tab)

Allows remote user full access to host's floppy drive.

Allows remote user read-only access to host's floppy drive.

Prohibits remote user from accessing host's floppy drive.

Allows remote user full access to host's fixed drive.

Allows remote user read-only access to host's fixed drive.

Prohibits remote user from accessing host's fixed drive.

Allows remote user full access to host's network drive.

Allows remote user read-only access to host's network drive.

Prohibits remote user from accessing host's network drive.

Allows remote user full access to host's CD-ROM drive.

Allows remote user read-only access to host's CD-ROM drive.

Prohibits remote user from accessing host's CD-ROM drive.

Remote Networking (Dial-Up Networking dialog box):

Check to establish remote control connection after remote networking session begins.

Click to specify options in remote control connection item.

Online connection item: Settings tab

Displays name of this online service connection item.

Sets type of terminal emulation to use during connection with online service.

Sets protocol to use during file transfer with online service.

Type phone number of online service.

Type phone number of online service.

Online Connection item: Session tab

Check to record online session in file you select. (If you check this option and do not enter a filename in the area below, you will be prompted for one when making a connection.)

Type name of file in which to record online session. (If you do not type a filename, you will be prompted for one when making a connection if you have checked the option above.)

Allows you to view and select existing files containing records of online sessions.

Enables you to run a script after connecting to online service.

Type name of script to run.

Allows you to view and select files containing existing scripts.

Enables use of translation table during online connection.

Type name of file containing translation table to use during online connection.

Allows you to view and select files containing translation tables.

Enables use of macro keys during online connection.

Type name of file containing configuration of macro keys.

Allows you to view and select files containing configurations of macro keys.

Saves statistics of online session in log file on this PC.

Records connection with online service as a script that can be re-used to repeat procedures.

Online connection items: Advanced tab:

Check to display lines that exceed the width of the terminal window on the next line.

Check so that when screen is filled, next line overwrites current top line when cursor reaches last position on the last line of the terminal display window.

Check to cause backspace key to act as a destructive, deleting key instead of a left cursor key.

Check to causes cursor to move to the left side of the terminal display window and to advance one line each time a carriage return character is received. (Check only if display does not scroll vertically as expected.)

Specifies length of the break signal used to interrupt programs running on a mainframe or minicomputer.

Click to restore original settings for the options in this dialog box.

Host Settings tab:

Displays name of this host connection item.

Sets time for delay before host calls back a remote caller, when callback is selected in that caller's connection item. (Disabled, if connection device other than modem has been selected.)

Determines who has use of keyboard and mouse during session.

This option is automatically set to Remote and cannot be changed if the Blank The PC Screen option is checked on the Security Options property page.

Causes the host to restart after every remote control session.

If checked, pcAnywhere displays minimized button on the task bar when host is waiting for remote PC to call.

Sets host PC to start pcAnywhere and begin waiting for a call as soon as Windows starts after boot-up. (If this host requires a network gateway to make connections, do not check this option.)

A connection that is unexpectedly ended.

A connection that is ended by the remote or host PC.

The number of minutes the host waits after an unexpected disconnect before returning to the selected end of session option.

The host is canceled and does not accept other connections.

The host returns to wait for another connection.

Check this option and select the action you want the host to perform after an unexpected loss of connection:

Logoff user: The host PC logs the user off the network. The host PC can accept another call, but the network login name and password is required access the network.

Restart host computer: The host PC reboots after the disconnect. If you want to have the host wait for another call, be sure to check the **Launch with Windows** option.

Use Windows screen saver: You can add a password to the host screen saver and select this option to prevent unauthorized users from viewing the host PC screen. An unexpected disconnect could leave confidential information on the host PC.

Check this option and select the action you want the host to perform after ending the remote control session.

Logoff user: The host PC logs the user off the network. The host PC can accept another call, but the network login name and password is required access the network.

Restart host computer: The host PC reboots after the disconnect. If you want to have the host wait for another call, be sure to check the **Launch with Windows** option.

Use Windows screen saver: You can add a password to the host screen saver and select this option to prevent unauthorized users from viewing the host PC screen. An unexpected disconnect could leave confidential information on the host PC.

Check Use directory services to choose the directory server the host waits on.

Logs the host user off the network preventing unauthorized access to the user's network privileges.

The host PC restarts.

 To have the host PC automatically wait for another call after restarting, check the Launch with Windows option.

Prevents unauthorized users from canceling the waiting host by locking the keyboard on the NT Workstation.

Prevents unauthorized users from canceling the waiting host by locking the keyboard on the NT Workstation.

The host is canceled and does not accept other connections.

The host returns to wait for another connection.

Logs the host user off the network preventing unauthorized access to the user's network privileges.

The host PC restarts.

 To have the host PC automatically wait for another call after restarting, check the Launch with Windows option.

Causes your windows 95 screen saver to launch whenever the host is waiting for a call. Add a password to your screen saver to lock the host and prevent unauthorized users from canceling the witing host.

Causes your windows 95 screen saver to launch whenever the host is waiting for a call. Add a password to your screen saver to lock the host and prevent unauthorized users from canceling the witing host.

Check to run the workstation as a Windows NT Service.

Host connection item Caller Tab:

Allows any caller to access this host with full privileges, no login names or passwords necessary.

Callers listed in pcAnywhere's Callers property page can connect using the login name, password, and privileges assigned by the **pcAnywhere host user**.

Callers configured on a Windows domain can connect using the login name, password, and privileges assigned by the **pcAnywhere host user**.

Callers configured on a Windows domain can connect using the Windows NT login name, password, and privileges assigned by the **network administrator**

Contains caller items with access to this host PC.

Host connection item NT Caller

Click to add a Windows NT user to the pcAnywhere caller list.

Click to add a Windows NT group to the pcAnywhere caller list.

If you have Windows NT administrator privileges you can use the Windows NT User Manager to add or delete users to the workstation or domain.

Host connection item, Security options tab:

Check to prevent use of host other than by remote control.

Check to prevent viewing of remote session at the host.

When a session unexpectedly ends, information could be left on the host screen. To prevent unauthorized callers from connecting and viewing this information, leave this option unchecked. Unchecked, the host accepts only the password of the remote caller that was connected at the time of the disconnect.

If left unchecked, passwords may be typed using any combination of uppercase and lowercase letters.

Check to log all connection attempts that do not succeed. Use as a security measure to track unauthorized connection attempts. (Applies only to connections with remote users for whom you create a caller item.)

Check to display, on connection, a prompt at the host that must be acknowledged before remote session continues. (Does not apply to callers who are Superusers.)

Set number of seconds that pcAnywhere waits for response by a host user when prompt on connection is enabled.

Check to have the host disconnect the session when the confirmation timeout expires.

Protects data being transferred by using one of three levels of data encryption available in the drop-down list box.

A list box containing three levels of encryption methods to use during remote control connections.

This options prevents a connection if the host and remote are not using the same level of encryption.

Provides the highest level of data security and is used when a certificate authority makes public keys available to the Cryptographic Service Provider on both host and remote sides of the session.

Select a private key from the list of currently installed private keys.

Type the name provided to the certificate authority at the time the certification was requested.

This option is case sensitive; type the name exactly as you provided it to the certificate authority.

Check to accept only calls from remote PCs using pcAnywhere for Windows, 2.0 and above, or pcAnywhere for DOS, version, 5.0, eliminating risk of data being transmitted without encryption.

Check to disconnect if login not successful within specified number of attempts.

Set maximum number of attempts at login before caller is disconnected.

Check to disconnect if login not successful within specified period of time.

Set time limit for login attempts before remote user is disconnected.

Check to disconnect if remote user is inactive for a specified period of time.

Set time limit for inactivity before remote user is disconnected.

HIDD Caller Folder textbox

Type folder to contain security settings for individual remote users calling this host.

HIDD Caller Folder Browse button

Allows you to view and select existing folders.

HIDD Settings button

Accesses dialog box with options to set for individual remote users.

Host connection item, Conference tab

Check if you want this host to allow multiple callers to connect and view activities on the host PC.

The first caller to connect remotely controls the host PC, while other callers connecting can only view the actions performed on the host PC.

Select to have the IP address automatically selected from any valid Class D addresses.

To increase the range of this host's broadcast, type in the number of routers to include.

To increase the range of this host's broadcast, type in the number of routers to include.

Select to manually provide an IP address for the conference. Use any IP address within the range of 225.1.1.1 through 239.254.254.254.

Greyed IP address and router boxes

Type a multicast IP address for this conference host PC. Type a range within 225.1.1.1 through 239.254.254.254.

Type a multicast IP address for this conference host PC. Type a range within 225.1.1.1 through 239.254.254.254.

Type a multicast IP address for this conference host PC. Type a range within 225.1.1.1 through 239.254.254.254.

Type a multicast IP address for this conference host PC. Type a range within 225.1.1.1 through 239.254.254.254.

Including additional routers increases the range of this host's broadcast.

Settings page for modems under nt

Speed of transmission of modem.

Sets error-checking procedure. Must be same for remote and host PCs.

Sets handshaking protocol.

Sets pcAnywhere to make connections through modem.

Selects type of modem to use.

Check to use modem connected to this port (instead of cable).

Choose modem connected to this port.

Click to set options for the modem connected to this port.

Advanced Modem Options page for modems under nt

Check for older rotary telephone systems that don't support tone dialing.

Check to instruct modem to use a leased telephone line rather than standard telephone service.

Check to match the serial port speed to the connect speed of the modem. Check this option if you selected Hayes Compatible and your modem supports speed adjustment.

Type the number of seconds pcAnywhere allows for a successful connection after a number has been dialed.

Type the number of seconds pcAnywhere allows for a successful connection after a number has been dialed.

Type the number of consecutive redial attempts that the pcAnywhere remote makes when a successful connection is not achieved. Type 0 to disable the redial feature.

Type the number of consecutive redial attempts that the pcAnywhere remote makes when a successful connection is not achieved. Type 0 to disable the redial feature.

Type the number of seconds a pcAnywhere remote pauses between redial attempts. *Enter 0 to allow unlimited time for connection after host dial.*

Type the number of seconds a pcAnywhere host pauses between redial attempts. *Enter 0 to allow unlimited time for connection after host dial.*

Type the number of rings after which the modem answers an incoming call. Type a value between 1 and 10.

Type optional AT modem commands to be added to the standard initialization string for your modem.

Type optional AT modem commands to be added to the standard initialization string for your modem.

Network device detail dialog box

Check to use a gateway PC to make remote connections outside of network.

Enter the name of a gateway you want to use when making a connection.

Enter the class of the gateway to view a list of all gateways with that class. Enter * to allow pcAnywhere to select any class.

Leave as “None” to use gateway’s parity setting.

Check to enter name of server in the textbox to the right.

Type name of asynchronous communications server. Leave as <Any> to allow pcAnywhere to select the first available server. Click the Browse button below to view a list of currently available servers.

Select to enter name of multiport serial card to use with the asynchronous communications server.

Enter name of multiport serial card to use with the ACS. Leave as <Any> to allow pcAnywhere to select the first available service. Click the Browse button below to view a list of currently available servers.

Select to enter name of port in text box to the right.

Enter name of port. Leave as <Any> to allow pcAnywhere to select the first available service. Click the Browse button below to view a list of currently available ports.

Select to allow remote caller to select the ACS server, service, and port at the time of connection.

Click to view a list of available servers, services, or ports.

Double-click to select server.

Application options -- Button Bars tab:

Check to place folder history tool on toolbar (when toolbar is visible).

Check to place browse folder tool on toolbar (when toolbar is visible).

Action Bar buttons appear as graphic symbols only, without text labels.

Action Bar buttons appear as graphic symbols and text labels describing their function.

Check to place Quick Start button on action bar.

Check to place Remote Control button on action bar.

Check to place File Transfer button on action bar.

Check to place Online Services button on action bar.

Check to place Be a Host button on action bar.

Check to place Gateway button on action bar.

Check to place Remote Networking button on action bar.

Check to place Exit button on action bar.

Application options -- Remote Operation tab:

Sets maximum number of colors pcAnywhere displays in remote control session. Faster performance results when fewer colors display.

If this option is checked, the pcAnywhere application remains open to allow the remote to call more than one host.

 To leave more memory available for your multiple sessions, leave this option unchecked. You can still quickly call multiple hosts by dragging your connection items to the desktop.

Specifies amount of disk space pcAnywhere uses to store information locally on the remote PC, thus increasing Windows' display speed. Space used only during remote session.

Check to have remote session take over full screen of remote PC.

If checked, host's screen image sized to fit within the remote PC's window so that remote's scroll bars and task bar remain available during session.

Check to improve performance by having remote PC use a local font that closely matches font used at host. If text is not correctly displayed on remote, uncheck this (using online menu).

Check to improve performance by disabling the host PC's wallpaper, screen saver, idle power-down options, and by preventing full window dragging. (All options are restored at end of session.)

 This feature disables a screensaver only if it appears in the Screensaver tab of Microsoft's Display Properties dialog box after installation.

Check to improve performance by disabling all Active Desktop settings on the host PC.

Specifies which keyboard handler to use during a session. Can be changed during a session.

Check to provide best possible representation of DOS graphics. Uncheck to improve speed.

Slows host application to the rate at which remote can display screen activity.

Click to determine the background and foreground colors to use during the session, as well as font style and size.

Check this option to lower the resolution on the host PC to match the resolution used on the remote PC.

This option can also be enabled during a session from the Online Options menu.

This option automatically moves the remote view to a currently active window on the host PC. For example, an error message on the host could appear out of the remote's viewing area, causing the remote caller to think the session has locked. If this option is checked, the remote view automatically displays the error message window allowing the remote caller to cancel the message.

Check this option to use the remote's mouse configuration on the host PC. For example, if the remote caller is using a left-hand mouse configuration, the host mouse adjusts to allow left-hand mouse control during the session.

 This option can also be enabled during a session from the Online Options menu.

Application options: Remote Printing tab:

To print to the remote's printer during a session, add the remote's printer definition.

You must fist install the remote's printer definition on the host PC using the Windows 95 or Windows NT control panel.

Lists currently defined remote printers.

Runs the Add Printer Wizard to create a new remote printer definition.

The remote's printer driver must be installed on the host PC before you can create a definition for it. If the Add Printer Wizard does not list the remote printer driver, install it following the manufacturers instructions.

Removes the selected printer from the list.

Application options: System Setup tab:

Choose to use Windows' computer name (determined at time of Windows' installation) to identify this PC for pcAnywhere.

Displays Windows' computer name.

Choose to name PC yourself for pcAnywhere sessions.

Type in name that will identify your PC in remote control sessions.

Click to enter information about your phone line and telephone number, calling card, and to disable call waiting.

Where you are dialing from. (Click Dialing Properties to change this.)

This is the area code you are calling *from*. If this area code is not correct, click Dialing Properties to edit or change your location.

A certificate store is created by the network administrator and contains certificate authorities. See your network administrator for the name of the Cryptographic certificate store.

Click Browse to locate the name of the certificate store. A certificate store is a database containing security certificates.

Check this option to use the Windows NT Event Viewer to monitor the server security events.

Application options -- TCP/IP tab:

When this PC is a remote, you can filter the list of available TCP/IP hosts by adding the name, IP address, or subnet address of the host PCs to search for.

A list of the TCP/IP hosts to search for.

Click to add the host to the search list.

Click to delete a TCP/IP host from the search list..
(You must select the host you want to remove first.)

When this PC is a host, you can restrict access to the host by adding the name, IP address, or subnet address of the remote PCs allowed to connect.

A list of the TCP/IP remotes allowed to connect to this host.

Click to add a pcAnywhere remote to the restricted remote list.

Click to remove a pcAnywhere remote from the restricted remote list.

Application options – Host Operation tab:

Change the default setting only if you experience video problems on the host PC.

Change the default setting only if you experience video problems on the host PC.

Check to display a warning dialog box whenever an unsecured host connection item is launched.

 An unsecured host connection item is not protected with caller authentication or caller password.

Application options: Terminal Emulation tab:

Allows dynamic selection of the fonts used based on the size of displaying window.

Displays vertical and horizontal scroll bars.

Displays terminal's status line.

Click to select file containing user-defined macro keys.

Click to select translation file, allowing advanced users to configure translation of incoming or outcoming characters.

Translation Table dialog box

Choose a code that you want to edit from this list.

Code you are editing.

Code as it is received.

Code as it is transmitted.

Macro Keys configuration

Displays list of possible macro keys.

Choose so that selected macro keys to the left perform a function that sends text.

Choose so that selected macro keys to the left perform a function that executes a script.

Choose so that selected macro keys to the left perform a function that runs a program.

Enter name of file (text, script, or executable program) assigned to macro keys selected above.

Application options: File Transfer

Displays file transfer protocols that can be customized. Choose the one you want to customize.

Click to customize options for file transfer protocol you have selected to the left.

For file transfer protocols that do not prompt you for a specific download folder before each transfer.

Click to view available folders for file transfer downloads.

Check to pause the remote control session during file transfers. If you want simultaneous remote control and file transfer functionality, leave this option unchecked. This option can also be enabled or disabled during a file transfer.

pcAnywhere protocol options dialog box (accessed through App Options, File Transfer, pcAnywhere settings):

Set the file overwrite option to use when a duplicate filename is found.

Check to compress files during transfer. Files are compressed during transfer to improve speed, but are automatically uncompressed to their original state before being written to the destination drive.
(Do not use for files that are already compressed.)

If file transfer is interrupted before completion, pcAnywhere will continue the tranfer at point of interruption when connection is re-established.

ASCII protocol customization (accessed through App Options, File Transfer, ASCII settings):

Type the number of tenths of seconds to pause after a character is sent.

Type the number of tenths of seconds to pause after a line is sent.

Enter the ASCII value of the character that causes transmission to pause until the character is received.

Type the maximum number of seconds of inactivity before transfer is terminated.

Select the way to handle a carriage return when sending files.

Select the way to handle a linefeed when sending files.

Select the way to handle a carriage return when receiving files.

Select the way to handle a linefeed when receiving files.

zmodem protocol customization (accessed through App Options, File Transfer, zmodem settings):

Check if you want 16-bit error checking.

Check if you want 32-bit error checking.

Check if you have a clean phone line.

Check if you have a noisy phone line.

Check if you have a noisy phone line and the 2K setting is not adequate.

When checked, if file transfer is interrupted before completion, you can reconnect and continue transferring the file from the point of interruption.

When checked, the online service can transfer files without initiation from the remote user.

Gateway connection item, Settings tab:

Displays name of gateway connection item.

If checked, gateway allows both incoming and outgoing calls. Leave unchecked to restrict gateway so that it allows only incoming or outgoing calls.

Specifies this gateway as part of a group of gateways, generally based on a shared modem speed.

Check to terminate sessions using this gateway after specified period of inactivity.

Specifies maximum period of inactivity allowed before gateway automatically ends an idle session.

Activity log processing db

Choose a type of log file to create reports on or to archive and/or delete.

Click to create report using information in the log files you've selected in the list to the left.

Click to archive and/or to delete the log files you've selected in the list to the left.

Host & Remote Session Log Report dialog box

Formats data in columns, with title and column headings. Includes page numbers.

Choose to create a report displaying information in columns.

Formats data for use in a database application or other program that requires delimiters between fields. Each item is enclosed in double quotes, followed by a comma. Month, date and year are separated by slashes (/). Each entry is followed by a carriage return.

Each item is entered in a field of fixed size.

Type the year of the earliest session date to include in the log report.

Type the month of the earliest session date to include in the log report.

Type the day of the earliest session date to include in the log report.

Type the year of the latest session date to include in the log report.

Type the month of the latest session date to include in the log report.

Type the day of the latest session date to include in the log report.

Check to include comments typed by user at end of session.

Check to include the date and time that sessions began.

Logs host name.

Check to include information on any file transfers during sessions.

Logs name of registered pcAnywhere host user.

Check to include name of host PC for each session included in the report.

Check to include date and time that sessions ended.

Logs name of the remote user.

Logs name of the remote computer.

Logs name of registered pcAnywhere remote user.

Check to include the date and time that sessions began.

Check to include date and time that sessions ended.

Logs reason for disconnection.

Archive/Delete Host/Remote Log dialog box

Type year of earliest log files you want to archive and/or delete.

Type month of earliest log files you want to archive and/or delete.

Type day of earliest log files you want to archive and/or delete.

Type year of latest log files you want to archive and/or delete.

Type month of latest log files you want to archive and/or delete.

Type day of latest log files you want to archive and/or delete.

Check to delete log files between the dates above from this PC.

Check to archive log files between the dates above.

Online DOS Settings tab

HIDD Online DOS Blink Attribute

Check to allow blinking characters.

Check to allow most accurate representation of DOS graphics possible. Uncheck to improve screen display speed of an approximation of graphics is available.

Check to slow host application to the rate at which the remote can display screen activity.

Select keyboard handler in order to allow execution of host applications that handle keyboard input in a nonstandard fashion.

Online General tab

Determines where printing during online session occurs.

Click to configure printing setup.

Check to prevent use of host PC's keyboard during online session.

Check to prevent display on host PC's screen during online session.

Check to remove scroll bars from display of host PC's screen, scale display to fit your monitor.

Check to display remote online toolbar.

Choose the number of colors in which the host PC's display appears. Selection of fewer colors provides faster performances.

Choose the number of colors in which the host PC's display appears. Selection of fewer colors provides faster performances.

HIDD Online Misc Full Screen Display

Check to ensure full-screen display during online session.

Transfer Clipboard db

Select to transfer host's clipboard to remote's clipboard.

Select to transfer remote control's clipboard to host's clipboard.

Check to transfer text in clipboard.

Check to transfer graphic in clipboard.

Containers, connection items

Begins Be a Host Quick Start Wizard. Double-click to create a new Be a Host connection item.

Connection item for this PC as a pcAnywhere host. Double-click to wait for a call.

Displays connection items for this PC to use when acting as a host. Right-click one to change its properties. Double-click to wait for a call.

Begins Remote PC Quick Start Wizard. Double-click to create a new Remote Control connection item.

Connection item for this PC as a remote PC. Double-click to make a connection.

Displays host PCs that this PC can control. Right-click one to change its properties. Double-click to call that host.

Displays host PCs with which this PC can transfer files. Right-click one to change its properties. Double-click to call that host and initiate file transfer session.

Begins File Transfer Quick Start Wizard. Double-click to create a new File Transfer connection item.

Connection item for this PC to begin file transfer session with host PC. Double-click to make a connection.

Begins New Caller Quick Start Wizard. Double-click to create a new individual caller.

Item containing login name, password, and security settings for an individual caller.

List of callers that can connect to this host PC. Right-click one to change its properties, including login name, password, and security options.

Begins Gateway Quick Start Wizard. Double-click to create a new Gateway connection item.

Double-click to activate this gateway PC.

List of connection items for this PC to use when acting as a gateway. Right-click one to change its properties. Double-click to activate gateway.

Begins Online Service Quick Start Wizard. Double-click to create a new Online Service item.

Connection item to call an online service. Double-click to call online service.

List of connection items this PC uses to call online services. Right-click one to change its properties. Double-click to call the service.

Begins Remote Networking Quick Start Wizard. Double-click to create a new Remote Networking connection item.

Connection item for Remote Networking. Double-click to make a Remote Networking connection.

List of connection items for this PC to use dialing in to a network. Right-click one to change its properties. Double-click dial in to a network.

System menus

HIDD Be a Host System Menu

Options available only during a connection.

HIDD Remote Control System Menu

Options available only during a connection.

NETBIOS control

You can enter the LAN Adapter Number (LANA) that is configured in the Windows 95 or Windows NT operating system

Scripts dialog box

Click to run a script you have selected in the list to the left.

Click to compile a script you have selected in the list to the left.

Click to edit a script you have selected in the list to the left.

Click to create a new script in the Script Editor.

Click to delete a script you have selected in the list to the left.

Displays list of existing scripts.

Displays folder containing scripts appearing in list above. (Click Browse to change folders.)

Click to view or browse other folders.

The following is the description for the Script Editor dialog box: To view: Tools/Scripts/New/Help

Menu F1 pop-ups

Creates new connection item in the current pcAnywhere mode.

Allows you to choose a printer and printing options.

Opens property sheets of selected connection item.

Opens property sheets to customize pcAnywhere.

Closes pcAnywhere.

Undoes last action.

Copies selected connection item to Clipboard.

Pastes connection item into current pcAnywhere window, if appropriate.

Changes pcAnywhere mode to Be a Gateway.

Changes pcAnywhere mode to Be a Host PC.

Accesses caller items.

Makes a connection using selected connection item.

Changes pcAnywhere mode to Call an Online Service.

HID_ACTION_CONTROLPANEL

Changes pcAnywhere mode to File Transfer.

HID_ACTION_LAUNCH

Activates gateway.

Sets host to wait for a call.

Presents QuickStart Wizards.

Refreshes connection items in current window. (Displays full names, etc.)

Provides context-sensitive help on the selected command or item.

Changes pcAnywhere mode to Remote Networking.

Changes pcAnywhere mode to Remote Control.

Starts file transfer session.

Waits for connection using selected connection item.

Automatically arranges connection items.

Arranges gateway connection items by their class.

Alphabetizes connection items in current window by connection device.

Alphabetizes connection items by terminal emulation. (Terminal emulation chosen in Settings tab of each item.)

Alphabetizes connection items by name of network host. (Host's name entered in Settings tab of each item.)

Alphabetizes connection items in current window.

Arranges connection items by telephone number of host. (Host's phone number entered in Settings tab of each item.)

Alphabetizes connection items by file transfer protocol. (Protocol chosen in Settings tab of each item.)

Alphabetizes host connection items in this window by caller folder (folder containing caller items for this host).

Accesses default application options.

Deletes selected connection item.

Creates log reports and archives log files.

Allows playback of recorded sessions.

Accesses Script Editor.

Converts configuration files from previous versions of pcAnywhere.

Displays all details of connection items in current mode.

Displays connection items in current mode as large items.

Displays connection items in current mode in list.

Displays connection items in current mode as small items.

Toggles display of action bar.

Toggles display of status bar.

Toggles display of toolbar.

Displays help file's table of contents.

Displays information about product's version.

Activates Quick Start Wizard to create a host connection item.

Activates Quick Start Wizard to create a remote control connection item.

Activates Quick Start Wizard to create a file transfer connection item.

Activates Quick Start Wizard to create a gateway connection item.

Activates Quick Start Wizard to create an online service connection item.

Administrative options - when pcAnywhere run with winaw32.exe /a

Check to allow display of folder tools on network users' when viewing this mode. Users can then select another folder in which to store and create their connection items.

Choose network folder in which to store pcAnywhere connection items for users. If option above is unchecked, this is the folder for all users' items.

Click to view and select other folders.

Type password to access administrator features.

Allows you to change the administrator password required to access and change network options of pcAnywhere.

Allows you to set location of connection items and network user privileges for all modes of pcAnywhere.

File manager window

A list of files in the selected folder on the Remote PC.

Click to change the drive on your PC displaying here.

A list of files in the selected folder on the Host PC.

Click to change drive on other PC displaying here.

Click to change direction of transfer.

Click to begin transfer.

Click to copy selected files to another folder on the same PC.

Click to delete selected files.

Click to rename selected file(s).

Adds the selected folders or files to an AutoTransfer procedure.

Click to create a new folder within the selected one.

Click to change folder viewed here.

Click to synchronize two folders displaying in windows above.

Displays information about your PC.

Displays information about other PC with which you are exchanging files.

Displays name identifying your PC in pcAnywhere sessions.

Displays name identifying the PC to which you are connected.

Playback Options dialog box

Check to manage playback from the Control Panel.

Choose the session to begin playing back.

Date of last recording.

Choose to stop playback after each session.

Choose to repeat the session you selected to start the playback.

Check to play all sessions and repeat them.

Choose speed of playback.

Playback Control Panel

Click to begin playback.

Click to reverse playback.

Click to fast-forward playback.

Click to stop playback in session.

Click to pause playback in session.

Click to ?

Choose speed of playback.

Click to save current screen of playback in a "screen shot".

Click to save session in a new file, either another record file or a raw binary file.

Group box names

Help is available for each item in this group. Point directly at an item and right click, then left-click the What's This option.

Help is available for each item in this group. Point directly at an item and right click, then left-click the What's This option.

Help is available for each item in this group. Point directly at an item and right click, then left-click the What's This option.

Help is available for each item in this group. Point directly at an item and right click, then left-click the What's This option.

Help is available for each item in this group. Point directly at an item and right click, then left-click the What's This option.

Display dialog box

If checked, pcAnywhere automatically selects the font size displayed on the remote screen, based upon the size of the terminal window.

If checked, horizontal and vertical scroll bars display.

Displays line at bottom of terminal window containing current terminal type, communication port, data rate, parity, and flow control.

Hardware settings dialog box

The speed at which information is moved from one location to another; measured in bits per second, or bps.

The speed at which information is moved from one location to another; measured in bits per second, or bps.

Sets error-checking procedure. Must be same for remote and online service.

Sets error-checking procedure. Must be same for remote and online service.

Sets handshaking protocol.

Sets handshaking protocol.

File Transfer protocol db.

A commonly used file transfer protocol that transfers one file at a time in blocks of 128 bytes. Do not use when transferring files with CompuServe or over asynchronous outdial networks like Tymnet.

A modified version of the original XMODEM with better error correction. Do not use when transferring files with CompuServe or over asynchronous outdial networks like Tymnet.

A modified version of the original XMODEM that transfers files in 1024 bytes. Do not use when transferring files with CompuServe or over asynchronous outdial networks like Tymnet.

A version of XMODEM 1K without error correction. Select only when both you and the Online Service are using error-correcting modems which use MNP or V.42 error correction. Do not use when transferring files with CompuServe or over asynchronous outdial networks like Tymnet.

Features high reliability, 1024 byte transfer packets and multiple file transfers with a single command (batch transfers).

Features 1024 byte transfer packets and multiple file transfers with a single command (batch transfers), without error correction. Select only when both you and the Online Service are using error-correcting modems which use MNP or V.42 error correction.

A very reliable feature packed file transfer protocol. Its extremely robust error correction make it an excellent choice where excessive line noise is a problem. Protocol options may be specified in the Settings Dialog group box.

Originally developed for communication between mainframe computers, this popular public domain protocol has been adapter for use on virtually every type of computer.

Refers to the transmission of plain text files using minimal error correction. Protocol options may be specified in the Settings Dialog group box. It is usually preferable to use a binary file transfer protocol (those listed above) if it is available.

Check to have any selected files downloaded automatically without initiation of a Receive command.

Choose settings for the ASCII protocol you've selected.

Choose settings for the ZMODEM protocol you've selected.

Pacing specifies a delay after each line or character that is sent.

Choose for this level of error correction.

Choose this option for highest level of error correction.

Enter the number of tenths of seconds to pause after a character is sent.

Enter the number of tenths of seconds to pause after a character is sent.

Enter the number of tenths of seconds to pause after a line is sent.

Enter the number of tenths of seconds to pause after a line is sent.

Enter the ASCII value of the character that causes transmission to pause until the character is received.

Enter the ASCII value of the character that causes transmission to pause until the character is received.

Choose error-checking options here.

Select None to halt transmission only when a negative acknowledgment (NAK) is received by the sending computer.

Use of a Data Window increases reliability, by pausing for an acknowledgment from the receiving computer after a specified amount of data has been sent. Select None to halt transmission only when a negative acknowledgment (NAK) is received by the sending computer.

download timeout

download timeout

Choose size of data window.

Choose size of data window.

When checked, if file transfer is interrupted before completion, you can reconnect and continue transferring the file from the point of interruption.

Select the way to handle a carriage return.

Select the way to handle a carriage return.

Select the way to handle a linefeed.

Select the way to handle a linefeed.

When uploading files (transferring files to the online service), specify how to treat these characters.

When downloading files (transferring files to the online service), specify how to treat these characters.

Select the way to handle a carriage return.

Select the way to handle a carriage return.

Select the way to handle a linefeed.

Select the way to handle a linefeed.

Search for Text dialog box

Type search text.

Choose to search upwards from current location in terminal window.

Choose to search downwards from current location in terminal window.

Check to find only text that matches upper/lower case as you typed it in the Search text area above.

Terminal settings db

Choose a different emulation, as specified by your online service.

Click to change the font, font style and point size used to display characters in terminal window.

Check to have the lines of text continue on the next line when the text exceeds the width of the display window. (If unchecked, the cursor remains on the same line until a line feed is received.)

Check to cause the Backspace key to delete characters as it passes over them.

Check to control what happens when text fills the entire screen.

Check if display does not scroll vertically as expected. pcAnywhere will advance one line when a carriage return is received.

Check to enable macro keys you have defined.

Check to enable a translation table.

File manager dialog boxes: Make folder

Displays folder on destination PC to which files will be sent.

Displays folder on PC from which files are transferred.

Choose or type folder you want to view.

Choose or type kind of file you want to select. (Example: ***.OSF**)

Type name of folder you want to create.

Displays name of currently displayed folder. Folder you create becomes a subfolder of this one.

Displays filename(s) of file(s) and folder(s) you have selected and are deleting.

Displays folder containing files you are deleting.

Check to delete any subfolders in folders you have selected for deletion.

Displays folder containing file(s) you are renaming.

Displays current name of file(s) you are renaming.

Type new name(s) of file you are renaming.

Displays folder containing file(s) you want to copy.

Displays name(s) of file(s) you want to copy.

Choose or type a folder where you want the file(s) above to be copied. (This folder must be on the same PC where the files are currently located.)

Displays name of source folder you want to synchronize.

Displays name of destination folder that you want to synchronize.

Choose this option to synchronize all files in the selected folders.

Choose this option to synchronize only the files you have selected.

Check to include subfolders in the synchronization.

Choose to view all files in the current folder.

Choose this option to view only program (executable) files in the current folder.

Choose this option to view only document files in the current folder.

Choose this option to filter the current folder to display only a group of files. Then choose or type the group (such as ***.OSF**) that you want to view.

To filter the current folder to display only a group of files, choose or type the group (such as ***.OSF**) that you want to view. (This selects the Custom option.)

Check to display subfolders in the current folder.

Displays source folder for cloning. Files in this folder will be duplicated in the destination folder on the other PC.

Displays destination folder for cloning. Files in this folder will be replaced by those in the source folder.

Check to include subfolders in the cloning procedure. (Files and folders currently located on the destination PC that are not on the source PC will be deleted!)

Choose an option to determine pcAnywhere's behavior when a file already exists on the destination PC.

- Never Overwrite Duplicate Files maintains original file on destination PC.

- Always Overwrite Duplicate Files automatically overwrites original file on destination PC.

- Verify Before Overwriting prompts you whether to overwrite original file. (Do not use for unattended file transfer.)

- Always Ask For Destination prompts you where to place new file transferred from source PC. (Do not use for unattended file transfer.)

- Overwrite Older Files Only overwrites file on destination PC only if its date is earlier than that on source PC.

Choose an option to determine pcAnywhere's behavior when a file already exists on the destination PC.

- Never Overwrite Duplicate Files maintains original file on destination PC.

- Always Overwrite Duplicate Files automatically overwrites original file on destination PC.

- Verify Before Overwriting prompts you whether to overwrite original file. (Do not use for unattended file transfer.)

- Always Ask For Destination prompts you where to place new file transferred from source PC. (Do not use for unattended file transfer.)

- Overwrite Older Files Only overwrites file on destination PC only if its date is earlier than that on source PC.

Check to compress files during transfer. Files are compressed during transfer to improve speed, but are automatically uncompressed to their original state before being written to the destination drive.

Check to allow reconnection and continued file transfer from the point at which file transfer was interrupted.

Check to turn on SpeedSend option, sending only portions of files that are different if those files already exist on the destination PC.

windows system key pressed dialog box

Choose to execute keystrokes on your PC.

Choose to execute keystrokes on the host PC.

Choose to ignore keystrokes.

not implemented.

ISDN dialog box

Check if you want pcAnywhere32 to attempt to combine two available 64K channels providing 128K per second transmission.

Enter Multiple Subscriber Number (MSN) extensions separated with a semi-colon, to restrict incoming calls to specific extensions.

AUTOTRANSFER pages

The AutoTransfer procedure filename.

Click to add new file transfer commands to the AutoTransfer procedure.

Deletes the selected command from the procedure.

Click to edit the selected command.

Select a command and click Move Up to place it in the order you want it to run. Commands are run sequentially, from top to bottom.

Select a command and click Move Down to place it in the order you want it to run. Commands are run sequentially, from top to bottom.

A list of the file transfer commands in the AutoTransfer procedure.

File or folder is sent from the remote PC to the host PC.

File or folder is sent from the host PC to the remote PC.

Checks duplicate filenames and transfers the file with the latest date and time, overwriting the older file.

If this AutoTransfer procedure is to run unattended, be sure to choose an overwrite options not requiring user confirmation.

The folder or file on the remote PC.

The folder or file on the host PC.

Disconnects the session when the AutoTransfer procedure ends. When this option is checked, file transfer errors are ignored and the procedure continues with the next command.

Scans downloaded files for viruses and deletes infected files.

Improves file transfer performance by checking duplicate files and transferring only the data that is different in the source file.

Check to override the file transfer options set in the Application Options, File Transfer property page. The settings are changed only in this AutoTransfer™ session.

Click to use the settings configured in the Application Options, File Transfer property page (displayed in grey).

Closes this dialog box and runs the selected procedure.

Click to view or modify the commands in this AutoTransfer procedure.

A list of the file transfer commands to be added to the procedure.

The name of the AutoTransfer procedure to run.

The name of the AutoTransfer procedure to add commands to.

Keyboard handlers help you to send special key sequences to the remote computer. Some applications, such as terminal emulation programs, require a keyboard handler to communicate directly with the host keyboard. Without this support, some or all of the keys in this type of program may be disabled in the remote session.

Select a keyboard handler level if you experience problems running DOS applications. Level one works well with most applications.

Slows the host application to the rate at which the remote can display screen activity.

Provides the best possible representation of DOS graphics. Uncheck to improve speed of session when an approximation of graphics is acceptable.

Determines background and foreground colors to use during the session as well as the font style and size.

The workstation or domain this user is is a member of.

The Windows NT user or group name of the selected caller.

The caller is an individual user with a name, password, and individual rights and privileges.

The caller is part of a Windows NT group and has all rights and privileges assigned to that group.

Logging Options

Check to record host sessions in a pcAnywhere recording file.

Type the full path to the file destination folder, or click Browse to select a destination folder.

If the file is stored to a central location, click Advanced to enter any authentication information required to access this location.

If the file is stored to a central location, enter the authentication information required to access this central location.

If the file is stored to a central location, enter the authentication information required to access this central location.

If the file is stored to a central location, enter the authentication information required to access this central location.

Check to use pcA Logging to log selected pcAnywhere events to a log file. You can choose to store this log file on the local PC or to a centralized server locati

A list of the pcAnywhere events you can choose to log.

Click to check all pcAnywhere events.

Click to uncheck all pcAnywhere events.

Reverses the current selection by *unchecking* all selected events and *checking* all unselected events.

Saves the pcAnywhere log file in the pcAnywhere Data folder on the local PC.

Saves the pcAnywhere log file to a destination on a central server.

Check to use SNMP event console and use SNMP traps to log pcAnywhere events.

A list of TCP/IP addresses to receive the pcAnywhere traps information.

Click to add additional TCP/IP addresses to the SNMP Trap Destinations.

Click to remove TCP/IP addresses from the SNMP Trap Destinations.

Check to log pcAnywhere events to the Windows NT event log.

DIRECTORY SERVICES

A list of the directory servers configured for this PC. Lists the display name of the directory server, the TCP/IP address of the server, and the verification status.

Click to add a new directory server to the list of available servers.

Removes a selected directory server from the list of server entries.

Click Properties to edit the configuration of a selected entry.

Use a descriptive name that clearly identifies the directory server. This name appears in the **Use Directory services** list box in the Settings property page of a remote control connection item.

Type the host name or the IP address of the directory server.

Click to set the Port Number the server uses to accept queries from the client and search base of the directory server.

Type the name and password used to access this directory server.

Type the name and password used to access this directory server.

The port that the directory server uses to accept queries from the client. The default port is 389.

The root of the directory structure where search queries begin.

DIAL UP NETWORKING

Click if you are connecting to a pcAnywhere network host on the internet.

Click if you are connecting to a pcAnywhere modem host on the internet.

Select a dial up networking configuration to use for this connection.

Click to add a new dial-up networking configuration.

Click Properties to edit the selected dial-up network configuration.

Type the user name and password used to access the host PC.

- If you are calling a Windows NT host, type the domain name

Type or select the number of redial attempts the pcAnywhere remote makes after an unsuccessful connection attempt.

Type or select the number of seconds pcAnywhere waits before attempting to redial the host.

Type the user name and password used to access the host PC.

- If you are calling a Windows NT host, type the domain name.

Type the user name and password used to access the host PC.

- If you are calling a Windows NT host, type the domain name

Check to include all files within subfolders in the file transfer.

Set the number of minutes to allow for inactivity before pcAnywhere disconnects the session.

Type the user name and password used to access the host PC.

- If you are calling a Windows NT host, type the domain name

Glossary Definitions

A

ACS (asynchronous communications server)

ANSI (American National Standards Institute)

asynchronous transmission

AutoTransfer

B

baud rate

BBS

blank

blink

bps

break length

break signal

C

cable

cache file

callback

caller authentication

carriage return

carrier

case-sensitivity

certificate

certificate authority

certificate store

channel

channel bonding

class

clone

communications

communications device

communications link

communications port (COM port)

communications protocol

communications system

communication session

compile

conference host

connection

connection device

connection item

drop-down menu

CR

crash recovery

Cryptographic Service Provider (CSP)

CTS

D

data bits

data conversion

data rate

data transfer

data transmission

destructive <backspace> key

Dial-Up Networking

direct connection

domain

DNS

DOS

DOS Host TSR

download

download folder

driver

DSR

DTR

E

error correction

F

file transfer

flow control

G

gateway

group (Windows NT)

O

online service

P, Q

parallel port

parity

parity bit

password

peripheral device

port

prefix

protocol

pulse dialing

R

RAS (Remote Access Server)

record

remote

Remote Access Server (RAS)

remote control session

remote communications

Remote Networking

remote terminal

resolution

router

RS-232-C

RTS

run

S

script

serial communication

serial interface

serial port

serial transmission

session

signal

signal state

smart terminal

SpeedSend

start bit

subnet address

stop bit

suffix

switched line

synchronize

T

TAPI devices

TCP/IP

terminal

terminal emulation

timeout

translation table

U, V ,W , X

upload

user account

user manager

user privileges

voice first

WAN

wildcard

wrap

XON/XOFF

ACS

A communications server that manages a pool of modems. It directs outgoing messages to the next available modem and directs incoming messages to the appropriate workstation.

ANSI

An organization of industry and business groups that develops voluntary U.S. standards for trade and communications. In pcAnywhere, this refers to terminal emulation that enables the use of ANSI commands (escape sequences) to control the screen and keyboard. These escape sequences have been standardized by ANSI.

asynchronous transmission

A way of transmitting data at irregular time intervals in which information is sent one character at a time. Each character contains a start bit, followed by a number of data bits and ending with a stop bit. This is the common method of communicating using a modem.

AutoTransfer

A file that contains commands to automatically transfer files to the host PC, receive files from the host PC, or synchronize files on both host and remote PCs. An AutoTransfer procedure can execute immediately upon connecting to the host PC.

baud rate

The number of times per second a signal changes in a communication channel. The term baud is often erroneously used to describe the speed at which the modem can transfer data. The correct measure for data rate is bits per second (bps).

BBS

Acronym for bulletin board system. A computer system that serves as an information and messaging center for a group of users who can dial in and connect using a modem and communications software. Online services often offer access to a BBS.

blank

Refers to a computer screen. A verg meaning to clear or not show an image on the screen. You can configure a pcAnywhere host to blank the host's screen once a connection has been made. This enhances the security of an unattended pcAnywhere host.

blink

Refers to a computer screen. The flashing on and off of a displayed character, such as a cursor. You can configure pcAnywhere to disable blinking characters if you do not want to support the blinking text attribute, or if it causes unstable display performance.

bps

Abbreviation for bits per second, a measure of the speed at which a device such as a modem can transfer data. Also known as data, or baud rate.

break length

In terminal emulation, this is the length of a break signal used to interrupt a program running on a mainframe or minicomputer. This is different from the CTRL-C or CTRL-BRK used by PCs. In pcAnywhere, you can specify the length of the break signal individually for each terminal emulation you are using.

break signal

A special signal used to interrupt whatever program is currently running and returns the user to the operating system, or some earlier level of a menu hierarchy.

cable

A group of wires enclosed in a protective tube, usually an organized set of wires that correspond to specific pins on a 9 or 25 pin connectors located at each end. A cable is used to connect peripheral devices to each other or to another computer. In remote computing this can refer to a cable used to connect a computer to a modem, or a cable that connects two computers directly, sometimes called a null modem cable.

Cache file

In pcAnywhere, a file used to improve the performance of Windows. The cache is established on the remote computer and is used to hold Windows' bitmap data. If the bitmap data is in the cache when a Windows screen is redrawn, the data does not have to be resent, which results in better performance.

Callback

In pcAnywhere, when a host disconnects a remote caller after a sucessful connection and then calls back the remote PC, either for security verification or financial responsibility.

Carriage return (CR)

A control character that tells a computer to return to the beginning of the current line. A separate character, the linfeed character (LF), advances the cursor to the next line. Because carriage return is often combined with linefeed, the combination is often referred to as carriage-return/linefeed, CR/LF or hard carriage return. In pcAnywhere, you can specify how the carriage return works for each terminal emulation.

carrier

The signal generated by a modem that is used to transmit data. The high-pitched sound (or ‘‘squeal’’) you hear when you initially connect with another modem is the carrier signal.

case-sensitivity

The discrimination between lowercase and uppercase characters.

certificate

A file used by cryptographic systems as proof of identity. It contains your name and your public key.

certificate authority

An office or bureau that issues security certificates.

certificate store

A database containing security certificates.

channel
In communications, a medium for transferring information, also known as a line or a circuit. Depending on its type, a communications channel can carry information in analog or digital form. A communications channel can be a physical link, such as the cable connecting two stations in a network, or it can consist of some electromagnetic transmission

channel bonding

For ISDN, channel bonding allows the two B channels to be combined into a 128 Kbps transmission. If second channel is available at the time PcAnywhere attempts to bond, the ISDN session will transmit at the 128K rate.

class
In pcAnywhere, a group of gateway hardware setups that have similar characteristics. The gateway hardware setup is assigned to a class when you create or edit the hardware setup. If a piece of hardware in a gateway cannot be accessed, pcAnywhere will use another piece of hardware with a hardware setup in the same class.
For example: You can create hardware setups for five 9600-bps modems attached to the network. If all 9600-bps modems are assigned to the same class, pcAnywhere will use one of the other modems if the specified modem is busy.

clone
In pcAnywhere, to make a specified folder on the host or remote PC identical to a specified folder on the other PC. Any files that are in the source folder, but not the destination, are copied to the destination folder. Files in the destination folder that are not in the source folder are deleted from the disk. Compare synchronize.

communications
The transfer of data between computers by means of a device such as a modem or a cable.

communications device or **connection device**
A modem, network interface card, or other hardware component enabling remote communications and data transfer between PCs.

communications link
A connection between computers (and/or peripherals) that enables data transfer. A communications link can be a network, modem or cable.

communications port (COM port)
A location for sending and receiving serial data transmissions. Also known as a serial port. These ports are referred to by the names COM1, COM2, COM3 and COM4.

communications protocol
A set of rules designed to allow computers to exchange data with one another with as little error as possible. Some protocols, such as RS-232-C, refer to hardware standards; others, such as XMODEM, refer to file transfer protocols. **See also** flow control.

communications system
The combination of hardware, software and data-transfer links that makes communications possible.

communications session
In communications, the time during which two computers (or a computer and a terminal) maintain a connection and, usually, are engaged in transferring information.

compile
The process of converting a high-level script into a low-level set of commands that can be executed, or run. Syntax errors are discovered when a script is being compiled.

conference host
A conference host allows multiple remote users to simultaneously connect and view the activities on the host PC. Only the first caller to connect can remotely control the host.

connection
The successful establishment of a communications link.

connection item
A host or remote connection item represents a pcAnywhere file containing connection device information and security settings to use during a session. To begin a session, double-click the connection item from a pcAnywhere window, from your Windows desktop, or from the Explorer.

Drop-down menu

The menu that appears when you right-click any connection item.

CR
See carriage return.

crash recovery
A file transfer option that directs pcAnywhere to continue transferring files where it left off when PCs are re-connected after a broken connection, instead of beginning the transfer all over again.

Cryptographic Service Provider (CSP)

An operating system software that provides cryptographic services compliant with the Microsoft CryptoAPI. Base-level CSPs are shipped with Windows NT 4.0 and Microsoft Internet Explorer 3.0 and higher.

CTS
Stands for Clear to Send. A signal sent from a modem to the computer to which it is connected, indicating that it is ready for transmission. CTS is sent over line five in standard RS-232-C connections.

data bits
In asynchronous transmission, this is the group of binary digits (bits) used to represent a single character of data. The number of data bits (usually 7 or 8 for modems) used in a transmission must be agreed upon by the sending and receiving computers. Each group of data bits in a transmission is preceded by a start bit and followed by an optional parity bit as well as one or more stop bits.

pcAnywhere uses 7 when parity is set to anything other than NONE.

See also asynchronous transmission, parity, start bits, stop bits.

Data conversion
In pcAnywhere, this feature allows you to convert configuration files (to connect to a host PC, for example) from an earlier version of pcAnywhere so that they can be used in the current version. You can also use this feature to import or export configuration files to text files for record-keeping purposes.

data rate
The speed at which information is moved from one location to another; measured in bits per second, or bps.
Compare baud rate.

data transfer
The movement of information from one location to another. The speed of transfer is called the data rate, or data transfer rate, and is usually measured in bits per second (bps).

data transmission
The electronic transfer of information from a sending device to a receiving device.

destructive <Backspace> key

When the Backspace key is pressed, characters are erased as the cursor moves to the left. A non-destructive Backspace key is identical in function to the PC's left arrow key.
pcAnywhere allows you to configure the Backspace key to function either as a delete key or as an arrow key, depending upon the type of terminal emulation you are using.

dial

In pcAnywhere, to initiate a connection via LAN, modem or direct connection, whether or not actual "dialing" is involved.

direct connection

A form of data communication where one computer or terminal is directly connected to another, usually via a null modem cable.

Domain

In Windows NT, a collection of computers defined by the administrator of a Windows NT server network that share a common directory database. A domain provides access to centralized user and group accounts maintained by the administrator.

Domain Naming System (DNS)

A hierarchical system of host naming that groups TCP/IP hosts into categories. For example, in the Internet naming scheme, names with extensions of "COM", identify hosts in commercial businesses.

DOS

DOS is the software that organized PC operations before the arrival of operating systems like Windows 95 and Windows NT. Programs that were created to run under DOS can run under Windows 95 and Windows NT, but they do not display the same interface elements as later programs do, such as windows, minimizing and maximizing buttons, and so on.

DOS Host TSR

In pcAnywhere, the DOS Host TSR is a terminate-and-stay-resident program which enables remote users to run on the Host, DOS applications full-screen within Windows or to exit Windows and run a program from the DOS command line. If this feature is enabled, the Host computer's AUTOEXEC.BAT file is modified to start the TSR at the time the computer is started.

download
In communications, the process of transferring a copy of a file from a distant or host computer to the local computer by means of a modem or network. In pcAnywhere, the process generally requires the remote PC to instruct the host PC to begin the transfer to the remote, which then stores the incoming file on disk. Compare upload.

download folder
The folder in which files received during file transfer are stored.

driver
Software instructions for interpreting commands for transfer to and from peripheral devices and the CPU.

DSR

Stands for Data Set Ready. A signal sent from a modem to the computer to which it is attached, indicating that it is ready to operate.

DTR

Stands for Data Terminal Ready. A signal used in serial communications that a modem sends to the PC to which it is attached, indicating that is available to accept incoming transmissions.

edit

To make changes to the content of a document or file.

encryption

A method or algorithm used to scramble data being transferred to protect the data from being intercepted by unauthorized persons.

error correction
Noisy lines or faulty connections can sometimes cause errors that translate into erroneous characters on the screen and in transferred files. Error correction causes data to be re-transmitted before it is displayed or transferred. pcAnywhere performs software-based error correction.

file transfer
The process of using communications to send a file from one computer to another. In communications, a protocol must be agreed upon by sending and receiving PCs before file transfer can take place.

flow control

A signal that acknowledges that communication or the transfer of information can take place. When a modem or computer receives data at a faster rate than it can process, data is stored in a special area of memory called a data buffer. Flow control, also called handshaking, prevents data loss by temporarily halting data transmission when the buffer approaches its capacity.

Handshakes can be controlled by either hardware or software. A hardware handshake, as between a computer and a modem, is an exchange of signals, over specific wires, in which each device signals its readiness to send or receive data (**see** RTS/CTS). A software handshake, usually exchanged during modem-to-modem types of communication, consists of actual information transmitted between the sending and receiving devices. A software handshake establishes agreement between devices on the protocols that both will use in communicating. **See also** protocol, XON/XOFF.

gateway
In pcAnywhere, a computer allowing users on a network to share a modem or other communications device when making connections. The gateway PC must have two devices. It takes the information from one communications device, converts it, and sends it out through the other.

A **bidirectional** gateway routes both incoming and outgoing calls.

group

In Windows NT User Manager, an account containing other accounts called members. Permissions and rights granted to a group are also provided to its members, making groups a convenient way to grant common capabilities to collections of user accounts.

handshake
A signal indicating that the communications devices involved are ready to communicate. Handshakes can be controlled by either hardware or software. A hardware handshake occurs when the two devices send signals over specific wires, indicating they are ready to send/receive data. A software handshake occurs when actual information is sent between the two devices. **See** flow control.

hardware setup
In pcAnywhere, a set of hardware parameters--such as modem type, port/device, data rate--that is used as a singular named resource in launching a host or remote session.

hexadecimal
The base-16 numbering system that consists of the digits 0 through 9 and the upper or lowercase letters A through F. Also called hex, hexadecimal is used in programming as a compact means of representing binary numbers used internally by the computer.

host
In pcAnywhere, this term refers to the PC that is controlled by the remote PC.

initialize
To prepare for use. In communications, to set a modem and software parameters at the start of a session.

IP address

Used to identify a workstation on a network and to specify routing information. Each workstation on the network must be assigned a unique IP address, which is made up of the network ID, plus a unique host ID assigned by the network administrator. This address is usually represented in dot-decimal notation, with the decimal value of each separated by a period (for example 123.45.6.24)..

IRQ
Interrupt ReQuests, also called hardware interrupts. They are the means that a connection device uses to signal other hardware components that it needs attention. When you start filling your PC with new devices (like serial ports, modems, and mice), you may find that previous devices no longer work.

Integrated Services Digital Network (ISDN)

A type of phone line used to enhance WAN speeds, ISDN lines can transmit at speeds of 64 or 128 kilobits per second, as opposed to standard phone lines, which transmit at only 9600 bits per second (bps). An ISDN line must be installed by the phone company at both the server and remote site.

keyboard handler
Keyboard handlers help you to send special key sequences to the remote computer. Some applications, such as terminal emulation programs, require a keyboard handler to communicate directly with the host keyboard. Without this support, some or all of the keys in this type of program may be disabled in the remote session.

In pcAnywhere, the Level 1 keyboard handler works well with most applications.

LAN
Acronym for local area network; a group of computers and other devices in a relatively limited area (such as a single building) and connected by a communications link that enables any device to interact with any other device on the network.

launch
To start a program or application. In pcAnywhere, the host PC is launched so that a remote PC can call it and begin a remote control session.

leased line
A telephone channel leased from a common carrier for private use. A leased line is faster and quieter than a switched line, but generally more expensive.

linefeed (LF)
A control character that tells a computer to advance one line below the current line. Because linefeed is often combined with carriage return, the combination is often referred to as carriage-return/linefeed or CR/LF.

log
In pcAnywhere, a record of activities and events that take place on a computer system during a remote control session. This record is stored in a data file.

login procedures
The process of identifying oneself to a computer after connecting to it over a communications line. During the login procedure, the computer usually requests the user's name and a password. On a computer normally used by many people, the login procedure provides a means of identifying authorized users, keeping track of their usage time, and maintaining security by controlling their access to sensitive files or actions.

pcAnywhere can be configured to require a password during login.

macro
A set of keystrokes and instructions recorded, saved and assigned to a short key code. When the key code is typed, the recorded keystrokes and instructions are executed (played back). Macros can simplify day-to-day operations that otherwise become tedious. For example, a single "macro" keystroke can set up a connection using pcAnywhere.

macro keys
Key codes assigned to sets of specific instructions. **See also** macro.

modem
A device that enables a computer to transmit information over a standard telephone line. Modems can transmit at different speeds or data transfer rates. **See also** baud rate, bps.

NACS
Stands for Novell's NetWare Asynchronous Communications Services. NACS allows any network PC to share dial-up phone lines or directly connected lines to a host or other asynchronous devices. In addition, remote users can access the network and transmit data over asynchronous protocols.

NASI
Stands for NetWare Asynchronous Services Interface. Network software used with Novell's NetWare Asynchronous Communications Services (NACS).

network
A group of PCs and associated devices that are connected by communications facilities (both hardware and software) for the purpose of sharing information and peripheral devices such as printers and modems. **See also** LAN.

network station
A computer connected to a LAN through a network adapter card and appropriate software.

null modem cable
A cable that enables two computers to communicate without the use of modems. A null modem cable accomplishes this by crossing the sending and receiving wires so that the wire used for transmitting by one device is used for receiving by the other and vice versa.

offhook
A telephone or modem that is in use.

onhook
A telephone or modem that is not in use.

online service
A computer communications system or service that allows users to dial in for information, messages and files. CompuServe is an example of a commercial online service.

parallel port
Transmits synchronous, high-speed flow of data along parallel lines. Usually used for printers.

parameters
A value assigned to a variable; in communications, a means of customizing program (software) and hardware operation.

parity
In communications, parity refers to an error-checking procedure in which the number of 1's must always be the same--either even or odd--for each group of bits transmitted without error.

In typical modem-to-modem communications, parity is one of the parameters that must be agreed upon by sending and receiving computers before transmission can take place.

Parity can be computed in any of the following ways:

Even: Adding the data bits and the parity bit yields an even number. If a character with an even number of bits arrives with the parity bit set (to 1), an error must have occurred during transmission.
Odd: Adding the data bits and the parity bit yields an odd number. If a character with an odd number of bits arrives with the parity bit set (to 1), an error must have occurred during transmission.
None: There is no parity bit.
Space: Sometimes a parity bit is used, but it is always set to 0. If a character is received with a parity bit set to 1, an error may have occurred during transmission. Space can also be used to transmit seven-bit characters to a device that is expecting eight-bit characters. Also referred to as **"**bit trimming".
Mark: Works the same way as Space, except that the parity bit is always set to 1. Also referred to as "bit forcing".

parity bit
An extra bit used in checking for errors in groups of data bits transferred within or between computer systems. In modem-to-modem communications, the sending computer adds a parity bit to each group of data bits, each of which represents a single character. The setting of the parity bit depends on the type of parity used. With even parity, for example, the parity bit is set to 1 whenever it is needed to make the total number of 1s (data bits plus parity bit) an even number. The receiving device counts the number of 1s in each arriving group of data and parity bits; if the number is odd when it should be even, the device can assume that one of the bits was transmitted incorrectly and that an error occurred.

password
A security measure used to restrict access to computer systems and sensitive files. A password is a unique string of characters that a user types as an identification code. The system compares the code against a stored list of authorized passwords and users. If the code is legitimate, the system allows access, at whatever security level has been approved for the owner of the password.

peripheral device
Piece of equipment that allows user to send and receive data to and from a PC, generally attaching to one of the PC's ports. Printers, modems, mice, and keyboards are all examples of peripheral devices.

port
A hardware location for passing data in and out of a computing device. Computers have ports for connecting peripheral devices, such as the COM (or RS-232-C) ports used to connect modems and printers. **See also** communications port.

prefix

In pcAnywhere, a code required before a telephone number (can be any number of digits). For example, the number 9 is often required to call out from many office PBX systems.

protocol
A set of rules designed to allow computers to exchange data with one another with as little error as possible. **See also** communications protocol.

pulse dialing
Rotary-style dialing (clicks can be heard when dialing) as opposed to touch tone.

record
To capture a chronological series of actions and events that occurred during a pcAnywhere session and store this information in a file. In a macro, keystrokes and program instructions are recorded for later.

remote
In pcAnywhere, a PC that connects with a host PC and takes control of it in a remote control session.

remote control session
Process in which a remote PC calls and connects with a host PC. The remote PC then operates the host while the host's video display is transmitted to the remote PC's monitor. CPU activity takes place on the host.
Compare Remote Networking.

remote communications
Interaction with a host by a remote computer through a telephone connection or another communications line, such as a network or a direct serial cable connection.

Remote Networking (RAS) or Dial-Up Networking
Connection in which a PC calls a network device and then operates as a node on that network. Compare remote control session.

Remote Networking is also referred to as Dial-Up Networking, Remote Access Service, or RAS.

remote terminal
A screen (monitor, video adapter and keyboard) that is located at a site removed from the computer to which it is attached. Remote terminals rely on modems and telephone lines to communicate with the host PC, or another communications link, such as a network or a direct serial cable connection.

resolution
A measure of the quality of an image, measured in pixels used to display alphanumeric characters of graphic images on the screen. High resolutions are composed of more dots per inch and appear smoother than low-resolution images.

In Windows 95, resolution is the Desktop Area in the Settings tab of the Display Properties dialog box for your desktop.

router

A device that helps Local Area Networks (LANs) and Wide Area Networks (WANs) achieve interoperability and connectivity.

RS-232-C standard
An industry standard for serial communication connections. Specific lines and signal characteristics are used to control the transmission of serial data between devices.

RTS
Request to send. An RS-232-C signal used in hardware flow control to pace information sent from one device to another. RTS is also used in most modems that equal or exceed 9600 baud, as well as direct connections.

See also CTS.

run
To execute a program or a script.

script
A type of program that consists of a set of instructions to an application. A script usually consists of instructions expressed using the application's rules and syntax, combined with simple control structures.

pcAnywhere source scripts have an extension of .SCR; compiled, executable pcAnywhere scripts have extensions of .SCX.

You can create a pcAnywhere script using the pcAnywhere Editor or any other standard ASCII text editor.

serial communication
The transmission of information between computers or between computers and peripheral devices one bit at a time over a single line (or data path 1 bit wide). Serial communications can be either synchronous or asynchronous. Both the sender and receiver must use the same baud rate, parity and flow control information. Most modems automatically synchronize to the highest baud rate both modems can support. pcAnywhere uses the asynchronous communications standard for personal computer serial communications.

serial interface
A data-transmission scheme that sends data and control bits in a 1 bit wide data path sequentially over a single transmission line. **See also** RS-232-C.

serial port
A location for sending and receiving serial data transmissions. Also known as a communications port or COM port. DOS references these ports by the names COM1, COM2, COM3 and COM4.

serial transmission
The transmission of discrete signals one after the other. In communications and data transfer, serial transmission involves sending information over a single wire one bit at a time; this is the method used in PCmodem-to-modem communications over telephone lines.

session
In communications, the time during which two PCs (or a computer and a terminal) maintain a connection and, usually, are engaged in transferring information.

signal
A general term for any electrical quantity that can be used to transmit or represent information.

signal state
A high or low voltage state. +3 volts or higher equals a binary 0, -3 volts or less equals a binary 1.

smart terminal
A PC running terminal emulation software that mimics a terminal, but is also capable of running a PC operating system, such as Windows ‘95. A pcAnywhere terminal is an example of a smart terminal.

SpeedSend

In pcAnywhere, an option that enhances file transfer performance when sending files with duplicate filenames by comparing the two files and transferring only the data that is different in the source file.

start bit
In asynchronous transmission, the bit that signals the beginning of a character. Start and stop bits are required in asynchronous transmissions because the irregular time gaps between transmitted characters makes it impossible for a receiving device to determine when the next character should arrive. These start and stop bits add considerable overhead to transmissions, increasing the transmission time as much as 20% over the synchronous equivalent. Usually, a transmitted character is made up of 1 start bit, 8 data bits and 1 stop bit.

stop bit
In asynchronous transmission, the bit that signals the end of a character. Start and stop bits are required in asynchronous transmissions because the irregular time gaps between transmitted characters makes it impossible for a receiving device to determine when the next character should arrive. pcAnywhere always uses 1 start bit and 1 stop bit.

suffix

In pcAnywhere, a code appended to the end of a telephone number, such as a calling card number for billing purposes**.**

subnet address

An address used to poll all 254 nodes on the designated network for pcAnywhere hosts. For example, an entry of 127.2.3.255 would display all pcAnywhere hosts with an IP address beginning with 127.2.3.

switched line
A standard dial-up telephone connection; the type of line established when a call is routed through a switching station. **Compare** leased line.

synchronize
In pcAnywhere, to copy files in both directions between two folders on host and remote PCs to make the folders identical to one another, each containing the same files. If two files with the same name are located, the file with the most current date and time is copied. Must be performed from the File Manager window. **Compare** clone.

synchronous transmission
A form of data transmission in which information is sent in blocks of bits separated by equal time intervals. The sending and receiving devices must first be set to interact with one another at precise intervals, then data is sent in a steady stream. **See also** asynchronous transmission (pcAnywhere uses asynchronous transmission).

syntax error
Syntax errors are detected during the compilation of a script and occur when you make a mistake entering a command, such as not enclosing a string in quotes, or specifying the wrong number of parameters. The syntax errors are written to a file with the same source filename and the extension .ERR. You can use the pcAnywhere Editor to view the .ERR file, make corrections to the script and attempt compilation again.

TAPI

An acronym for the Telephony Application Programming Interface available in Microsoft's Windows 95 and Windows NT, 4.0 operating systems. This feature automatically detects and configures communication hardware, including modems, installed on the PC.

TCP/IP
A common set of protocols used to link dissimilar computers across many kinds of networks. Used on the Internet.

terminal
A device consisting of a monitor, video adapter, and keyboard. A terminal does little or no processing on its own (also known as a dumb terminal); instead, it is connected to a computer with a communications link over a cable. Terminals are used mainly with multi-user systems, where they are used to monitor and receive but not store information (i.e., with a mainframe). **See also** terminal emulation, smart terminal.

terminal emulation
The technique of imitating a terminal by using software that conforms to a standard such as the ANSI standard. pcAnywhere can make your computer act as if it were a particular type of terminal in order to communicate with another computer, such as a mainframe or minicomputer.

timeout
A predetermined period of time during which a given task must be completed. If the timeout value is reached before or during the execution of the task, the task is canceled. You can configure a pcAnywhere host to disconnect from a remote PC after a certain amount time has passed without activity.

translation table
A code that allows data to be converted from one format to another.

voice first

Allows the host and remote user to have a voice conversation before beginning the data session. Use voice first when you have only one phone line and want to speak with the other user before starting the session.

upload
In communications, the process of sending a file from a one computer to a another by means of a modem, network or serial cable. With a modem-based communications link, the process generally involves the requesting computer instructing the distant computer to prepare to receive the file on its disk and then wait for the transmission to begin. **Compare** download.

user account

A Windows NT file containing information that defines a user to windows NT. This includes user name and password, the groups in which the user account has membership, and the rights and permissions the user has for using the system and accessing its resources.

User Manager

A Windows NT utility that allows a user with administrative privileges to edit and define individual user accounts and privileges for the local workstation.

user privileges

One of three privilege levels you can assign to a Windows NT user account. pcAnywhere uses the same privileges to configure a remote caller in the pcAnywhere Callers property page when the Use NT User Privileges option is selected.

WAN or Wide Area Network
Contrast with LAN. This network uses links provided by local telephone companies and usually connects disperse sites. Typically, it extends a local area network outside the building to link to other LANS in remote buildings, possibly in remote cities.

wildcard
A symbol that enables multiple matching values to be returned based on a shared feature. The script language has two wildcards: the question mark (?) and the asterisk (*). The question mark stands for any single character, and the asterisk stands for any character string of any length. For example, the file specification *.* would return all files, regardless of their filenames; the file specification *.SC? would return all filenames having a three-character extension beginning with SC (COMPUSRV.SCR, COMPUSRV.SCX, etc.).

wrap
The ability of the program to continue displaying information on a new line or page when the end of that line or page is reached.

workstation

Any networked computer using server resources.

XON/XOFF
The most common of protocols established to govern software handshaking. Under this protocol, the receiving device sends a specific character when it wants the transmitting device to stop sending characters. It sends a different character when it wants the transmission to resume.

Caller Authentication

In pcAnywhere, a caller's identity can be verified by matching the caller login and password to a Windows caller authentication list or a pcAnywhere caller list.

pcAnywhere modes

pcAnywhere has five modes of operation. Choose one from pcAnywhere's Action Bar to use connection items in that mode.

1 Be A Host — allows another PC to connect and operate yours.

2 Remote Control— to connect to and use a host PC.

3 File Transfer— to send or receives files from a host PC.

4 Gateway—allows other network PCs to use this PC's connection devices.

5 Call Online Service — allows connections to online services.

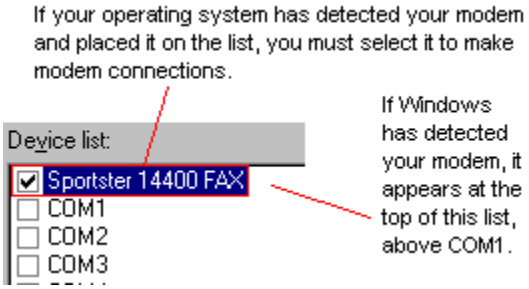
To include Online Service on the Action Bar choose Application Options from the Tools menu and click the Button Bar tab.

Connection Info tab

Check a connection device for remote control sessions using this connection item.

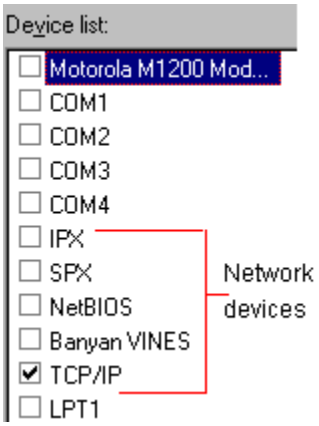
To find your modem:

Your operating system may detect any modem(s) connected to your PC. pcAnywhere then places the device at the top of the Device list:



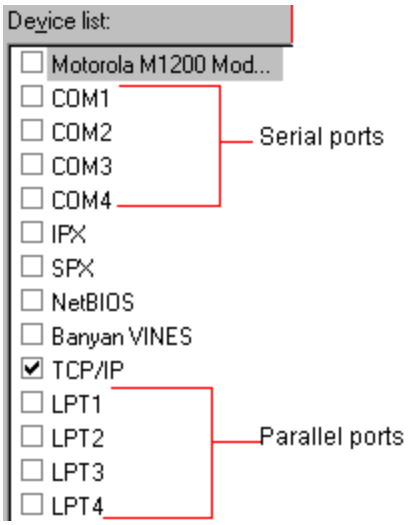
To find your network device:

Network devices appear in the middle of the list:



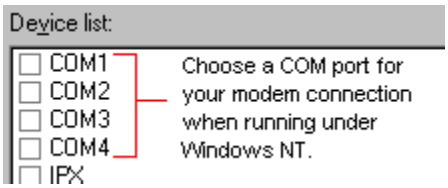
To find your cable device:

You must choose a COM (serial) port or LPT (parallel) port. You can customize the settings of COM ports.



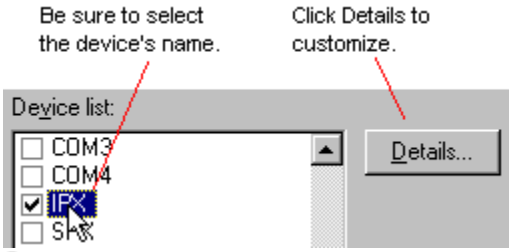
To find your modem under Windows NT:

If you are using Windows NT, choose the COM port connecting your modem to the PC, then customize it to choose your modem from pcAnywhere's list:



To customize a device:

- Click Details:



- For connection items in the Remote Control, and File Transfer modes: you must customize network devices if you want them to use a pcAnywhere gateway when making connections.

Connection Info

To choose the devices of a gateway:

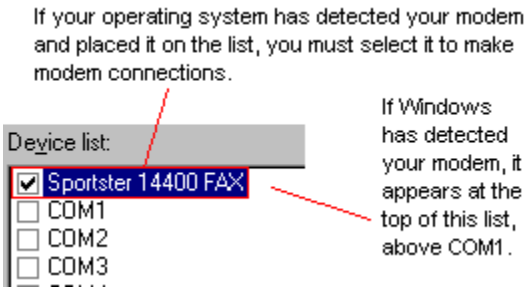
You must check two connection devices, one under Incoming device, and one under Outgoing device. Both devices of a bidirectional gateway can send and receive calls.

- You can make this a bidirectional gateway by checking the Bidirectional option on the Settings tab.

When choosing devices for a unidirectional gateway, however, set the incoming device as the one that connects to remote PCs. The outgoing device connects to host PCs.

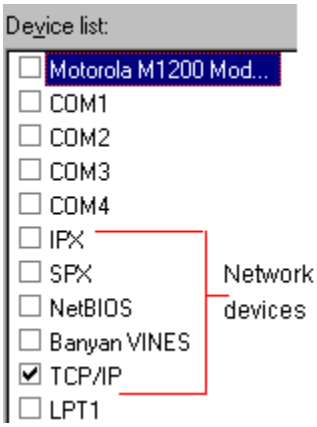
To find your modem:

If your operating system automatically detected the modem connected to your PC. pcAnywhere places that device at the top of the Device list:



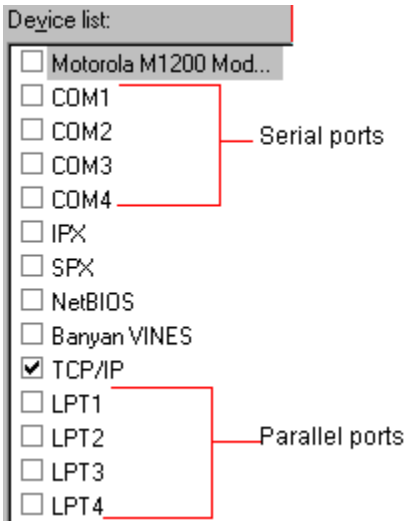
To find your network device:

Network devices appear in the middle of the list:



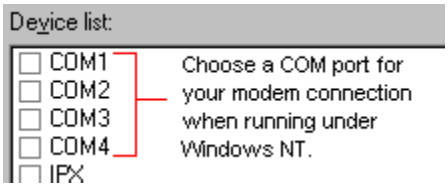
To find your cable device:

You must choose a COM (serial) port or LPT (parallel) port. You can customize the settings of COM ports.



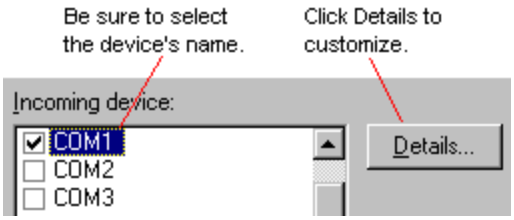
To find your modem under Windows NT:

If you are using Windows NT, choose the COM port connecting your modem to the PC, then customize it to choose your modem from pcAnywhere's list:



To customize a device:

- Click Details:



Host - Protect Item tab

Type a password to protect the use of this connection item.

- The password displays as asterisks to protect the security of the connection item. You may want to keep a record of what you typed in a secure location; forgotten passwords cannot be retrieved in pcAnywhere.

The password can restrict any or all of the following:

Required to view properties—check to require a user to enter the password before viewing any of the property pages for this connection item.

Required to execute—check to require a user to enter the password before using this connection item to make a connection or wait for a call. (If this is a gateway connection item, users must enter the password before activating the gateway.)

Required to modify properties—check to require a user to enter the password before changing any of the properties of this connection item.

Settings tab

Network connections:

Network host PC to control - type the host PC name in the text box.

If your connection device is TCP/IP, type the pcAnywhere host name, DNS name, IP address , or subnet address of the TCP/IP host to which you want to connect.

Use directory services and select a directory server to use for looking up a host

- The directory service option is greyed and not available if the network administrator has not configured this PC on a directory server.

Modem connections:

Use dialing properties and phone number - click to use the dialing information in the My Locations property page with the number when dialing the host.

Dialing Properties - click to view your current location's dialing properties. You can choose a different location or create a new location from this property page.

Use manually entered prefix, area code, phone number- click to type the prefix, area code, and phone number to dial and ignore the settings found in the My Locations property page.

- You type either the host PC's name or the phone number of the host PC, depending on what type of connection device you selected. To change the type of connection device, choose the Connection Info tab.

Select the following login information:

Automatically Login To Host Upon Connection - check to send your login name and password to the host automatically upon connecting.

- If you do not enter login information on this property page, you are prompted for it when making the connection.
- If you are calling a Windows NT host, you can include the domain name in addition to the login name. For example, type *domain\username* to automatically locate the user on a specific domain

Select the following connection options:

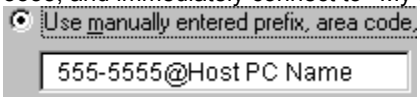
Number of connection attempts - type the number of times pcAnywhere will attempt to dial the host after a failed call.

Seconds between redials - type the number of seconds for pcAnywhere to wait before attempting to redial the host after a failed call.

For gatewayglossary_gateway **connections:**

- If you are using a gateway on your network to dial a host, you can type both the name of a host PC and a phone number. The gateway PC dials the number and connects you to the host.

If you are calling a host on an outside network that uses a gateway to make connections, type the phone number of the gateway followed by @ and the name of the host PC. For instance, to dial into a gateway at phone number 555-5555, and immediately connect to "My Host PC", you would type:



Automated Tasks

Select options to use whenever making a connection with this connection item.

To run an AutoTransfer procedure in all sessions with this connection item:

1 Check **Run upon connection** and click AutoTransfer:

2 Optional: Click Browse to select a different procedure to run

To run a script in all sessions with this connection item:

1 Check Run upon connection and click Script:

2 Click Browse.

3 Choose a script from the Script dialog box.

4 Click Select.

To record all sessions with this connection item:

1 Check Record session in file for later playback:

2 Optional: Click Browse and select a folder in which to place the file.

3 Type the name of the file in which to record the session.

- If you do not type a filename, you are prompted for one when making a connection if you have checked the option in step 1.

To create activity logs of all sessions with this connection item:

- Check **Save session statistics in activity log file**.

- You can run a script or begin recording at any point during a remote control session.

Settings tab

Options set on this tab affect all remote control sessions with this host PC:

Launch With Windows: Causes the host connection item to run and wait for a call automatically each time you start your PC.

Use Windows Screen Saver: Causes your Windows 95 screen saver to run whenever the host is waiting for a call.

- Add a password to your screen saver to lock the host and prevent unauthorized users from canceling the waiting host.

Lock NT Workstation: Prevents unauthorized users from canceling the waiting host by locking the NT Workstation with a password.

Run Minimized: Causes the host to appear on the desktop as an icon when it is waiting for a connection.

Run as a Service: Configures the host to run as an NT Service.

- The Run as a Service option is required if you want to launch the host with Windows.

Callback Delay: Causes the host to delay the specified number of seconds before attempting to call back the remote caller.

Abnormal Loss of Connection and End of Session Options: Allows you to determine the mode the host returns to after an unexpected disconnect or an end of session command.

- **Wait for Anyone:** Causes the host to wait for another call.
- **Cancel host:** Cancels the host. Other connections are not possible when the host is canceled
- **Secure by**

  Check to select any of the following actions

  - § **Logoff User**: Causes the host to log the current host user off the network, preventing unauthorized access to the user's network privileges
  - § **Restart Host Computer**: Causes the host PC to restart. If you want the host PC to wait for another call after restarting, check the Launch with Windows option
  - § **Use Windows Screen Saver**: Causes your Windows 95 screen saver to run whenever the host is waiting for a call.

Use directory services Allows you to select a directory server to use for locating a pcAnywhere host.

- The directory service option is greyed and not available if the network administrator has not configured this PC on a directory server.

pcAnywhere host - Callers property page

Choose one of the following options to verify and authorize callers connecting to the host PC:

Allow full access to all callers: All callers can connect to this host PC without login names or passwords, and with full remote control privileges

Use pcAnywhere authentication with pcAnywhere privileges: Callers listed in the Callers property page can connect using login name, password, and privileges assigned by the host PC user

Use Windows authentication with pcAnywhere privileges: Callers configured on a Windows domain can connect using the login name, password, and privileges assigned by the host PC user

Note: To use this option, the network administrator must create a share called "pcAnywhere$" on the primary domain controller.

Use Windows authentication with Windows privileges: Callers configured on a Windows domain can connect using the Windows NT login name, password, and privileges assigned by the network administrator.

The Caller items in the Caller property page window are like a directory listing of remote users who can call this host. Use the toolbar to create, modify, delete, etc. the caller items. Right-click any caller item and choose Properties to set that caller's privileges, login name, and password.

Each host connection item can be accessed by caller items in only one folder on the host PC. Change the caller folder to access or set up a different group of callers for this host.

To add remote callers and use Windows NT security privileges:

Select individual users or groups from the Windows NT user list. The users or groups on this list are created and maintained by the Windows NT administrator.

If you have administrator privileges on Windows NT, click NT User Manager to add or delete users.

Host Security Options tab

- Check these general security options for this host PC:

Blank PC screen after connection: blanks the host screen after connecting to prevent unauthorized viewing of session activities. Use this option if you remotely access confidential applications, such as email. Blanking the host screen during the session prevents others at the host site from viewing the contents of the application.

Prompt to confirm connection: alerts the host user about an incoming call and waits for a confirmation from the user before allowing the connection.

- **Timeout**: sets the time allows for the host user to respond to the confirmation prompt.
- **Disconnect if timeout**: disconnects the connection after the selected timeout elapses.

Make passwords case sensitive: requires a remote caller to type their password using the exact case of the password.

Limit login attempts per call: limits the number of times a remote caller can enter a login name and password.

Limit time to complete login: limits the amount of time the remote caller has to successfully login in the the host PC.

Encryption level: determines the level of data encryption this host uses during remote control sessions.

- **Deny Lower Encryption**: prevents a connection from a remote PC that is using a lower level of encryption than is set on the host.

Private Key Container: provides the highest level of data encryption and is used when a certificate authority makes public keys available to the Cryptographic Service Provider on both the host and remote sides of the session.

Certificate Common Name: contains the name provided to the certificate authority at the time the certificate was requested. The name is case sensitive and must be typed exactly as it was provided to the certificate authority.

Keyboard/Mouse active during session: determines who has use of the keyboard or mouse during a session. The default setting host and remote user. Set this to remote only to prevent users at the host site from using the host PC and inadvertently canceling the host.

Disconnect if inactive: disconnects the session if there is no activity detected on the host PC for the time specified in the Timeout box.

Settings tab

Terminal emulation: Choose the terminal emulation that the online service uses.

File transfer protocol: Choose the online service's file transfer protocol.

Use dialing properties and phone number: Select this option to use the current dialing properties and type the area code, phone number, and country code of the online service:

- Click Dialing Properties to change your dialing location or modify an existing location.

Number of redial attempts: Type the number of consecutive redial attempts the pcAnywhere remote makes when a successful connection is not made. Type 0 to disable the redial feature.

Seconds between redials: Type the number of seconds the remote waits before redialing the online service.

Some options affecting online service sessions are set by choosing either the Terminal Emulation tab or the File Transfer tab from the Application Options property sheets.

- You can change the protocol or terminal emulation used during an online service using options in the Terminal window.

Session tab

Select the pcAnywhere options to use whenever making a connection to the online service.

To record all sessions with this connection item:

1 Check Record session in file for later playback:

2 Optional: Click Browse and select a folder in which to place the file.

3 Type the name of the file in which to record the session.

- If you do not type a filename, you will be prompted for one when making a connection if you have checked the option in step 1.

To run a script in all sessions with this connection item:

1 Check Run script file after connect:

2 Type the name of the script file to run or click Browse.to locate a different script file.

To use a translation table during sessions:

1 Check **Use translation table file**:

2 Type the name of the translation table to use or click Browse to locate a different translation table.

To use macro keys during sessions:

1 Check Use Macro Key file:

2 Type the name of the macro key file or click Browse to locate a different macro key file.

To record sessions for use as scripts:

- Check Begin script recording after connection: to record the session to a recording file for later playback.

Advanced tab

The options you set here customize the terminal emulation you chose on the Settings tab for sessions with the online service.

Set these options:

- **Line wrap.** Check to display lines that exceed the width of the terminal window on the next line.
- **Screen wrap.** Check so that when screen is filled, next line overwrites current top line when cursor reaches last position on the last line of the terminal display window.
- **Destructive backspace key.** Check to cause backspace key to act as a destructive, deleting key instead of a left cursor key.
- **Translate Receive CR to CR/LF.** Check to causes cursor to move to the left side of the terminal display window and to advance one line each time a carriage return character is received. (Check only if display does not scroll vertically as expected.)
- **Break Length.** Specifies length of the break signal used to interrupt programs running on a mainframe or minicomputer.
- Click **Defaults** to re-set the options on this page as the default options for the chosen terminal emulation.

Gateway Settings

The gateway name is displayed in the properties page title bar. This is the name that users must specify when customizing a connection device to connect using this particular gateway. To change it, you must rename this connection item.

- Set these options:
- Bidirectional. Check to allow incoming and outgoing calls on either device selected on the Connection Info tab for this gateway.
- Inactivity timeout. Check to disconnect any session that remains inactive longer than time limit you specify.
- Class. Type name. (Places this gateway in a group of other gateways on a network. Users can specify the use of any gateway in a particular class when customizing connection devices.)

Caller Identification tab

Login name and password: Type a unique login name and password for each caller. This protects the host from unauthorized access.

To make the entry of passwords case-sensitive, check the **Make passwords case sensitive** option located in the Security Options tab of the host connection item.

- Each caller can automatically send the assigned login name and password to the host at connection time. To do this, remote users must enter this information in the Settings properlty page of the remote control conneciton item.

Callback tab

Call back the remote user: Check: to instruct pcAnywhere to terminate a connection (after the remote user has successfully logged in) and call back the remote PC.

Type the number of the remote PC in the text box or leave blank to prompt caller for number at time of initial connection. (User will not be prompted until after successful login.)

Callback can occur only on a host PC that makes modem connections.

- Use this feature to reverse telephone charges and as an additional security measure. If the callback phone number is provided in the Phone number text box, the host PC can verify the location and identify PC.

Caller Privileges tab

Click **Superuser**: to give the caller full access rights to the host.:

Click **Specificy individual caller rights**: to choose any of the following privileges for this caller:

- **Allow caller to blank screen**: check to allow the remote caller to blank the host screen after connecting. This option prevents unauthorized users from viewing the activities of the session.
- **Allow caller to cancel host**: check to allow this caller to cancel the host after ending the session. Canceling the waiting host prevents other callers from connecting.
- **Allow caller to restart host**: check to allow this caller the right to restart the host after ending the session.

 Restarting the host could prevent other connections if the Launch with Windows option is not selected in the host's Settings property page.

- **Allow caller to upload files**: Check to allow this caller to transfer files *to* the host PC.
- **Allow caller to download files**: Check to allow this caller to transfer files *from* the host PC.
- **Allow use of Ctrl+Break**: Check to allow this caller to use Ctrl+Break to terminate processes on the host PC.
- **Limit time allows per session**: Set the number of minutes to allow this caller to remain connected to the host PC.
- **Caller subject to inactivity timeout**: Check to terminate the session with this caller if there is no activity on the host for the Inactivity time limit set in the host connection item's Security Options property page.
- **Set Drive Access**: Click to set drive access privileges for this caller.
- **Command to execute after connect**: Type the name of a script file or automated procedure to run after this caller connects to the host.

Button Bars

Choose the appearance of the pcAnywhere action bar and toolbar.

Display these Action Buttons

The pcAnywhere Action Bar

The action bar is a graphical representation of the pcAnywhere functions listed in the View menu. You can customize the actions displayed on the action bar as well as the appearance of the button.

Check the action buttons you want on the pcAnywhere Action Bar and listed in the View menu.

- Click **Icon only** to remove the text on the button.
- Click **Icon and text** to display both the icon and the text on the

The pcAnywhere toolbar

Check **Allow folder browse**: to place the browse folder icon  on the toolbar to allow the user to browse folders.

Check **Show folder history**: to display a list of recently accessed folders in the drop-down text box.

Remote Operation

These Remote Operation options affect all remote control sessions for this PC.

Performance options

ColorScale: increases screen refresh performance by reducing the number of colors during a remote control session.

- The host PC's color palette must be set to 256 colors or greater for ColorScale to take effect during sessions with the host.

pcAnywhere cachefile size: specifies the amount of disk space pcAnywhere uses when caching information during a remote control session with PCs running Windows. This option increases the display speed of a session.

- If you are connecting to more than one host, you may want to reduce the cachefile size to ensure that cache is available for all sessions. pcAnywhere attempts to use the cachefile size for all sessions. If disk space is not available, the cachefile may affect the first session only.

Allow connection to multiple hosts: causes the pcAnywhere application to remain open, allowing the user to conveniently make another connection. If this option is not selected, multiple connections are made by restarting pcAnywhere.

- You can close pcAnywhere and still make multiple connections by dragging the connection items to the desktop. Double-click the desktop icon to start the connection.

Optimize Desktop For Remote Control—improves performance by disabling the host PC's wallpaper, screen save, idel power-down options, and by preventing full window dragging.

Disable Host's Active Desktop—improves performance by disabling any active desktop settings configured on the host PC.

Reduce Host desktop Area to Match Remote—improves performance by causing the host to reduce it's screen resolution to match the resolution used by the remote PC.

Navigation options

Screen scaling: scales the host screen image to fit within the remote PC's window so that the remote's scroll bars and task bar remain available during the session.

Local full screen display: displays the host image full-screen on the remote PC.

Host active window tracking: moves the remote view to a currently active window on the host PC. For example, an error message on the host could appear out of the remote's viewing area, causing the remote caller to think the session has locked. When this option is checked, the remote view automatically displays the error message window allowing the remote caller to cancel the message.

Use local fonts for Win 3.x hosts: improves the performance of the session by having the remote use a local font that closely matches the font used at the host. If text is not displayed correctly, you can uncheck this option from the online menu.

System Setup tab

Set default options to identify your PC and define its location for remote communications.

Your computer name selection

User-defined : Choose this option and type in the name that identifies this PC in pcAnywhere sessions.

Windows computer name: Choose this option to use the name of your PC as it was entered during Windows installation.

Dialing properties

Location: Displays the name of the location you are currently dialing from. This location is configured in the Dialing Properties dialog box.

Area code: Displays the area code you are currently calling from. This area code is configured in the Dialing Properties dialog box.

Dialing Properties: Click to view Microsoft's Dialing Properties dialog box and enter details about your communications device. Changes text in Location above.

Cryptographic certificate store: Type the name of the Cryptographic certificate store to use for encryption. A certificate store is created by the network administrator and contains certificate authorities.

Use NT Event Log (Windows NT only): Check this option if you want to log pcAnywhere session events to the Windows NT Event Log. Information recorded in this log include the date and time the event was recorded, the event category and ID number, the pcAnywhere user, and the name of the PC connecting.

TCP/IP Tab

Use the TCP/IP host list to add the names or addresses of pcAnywhere hosts you want to search for on your TCP/IP network or the Internet. The information on this list is used to generate an Internet SmartList of pcAnywhere hosts on the Internet. If this list is left blank, pcAnywhere lists TCP/IP hosts on the local network only.

TCP/IP hosts to connect to:

When this PC is a remote you can filter the list of available hosts by typing the pcAnywhere host name,DNSname, IP address, or subnet address of the TCP/IP host to search for.

Click **Add Search - ** to add the TCP/IP address to the list of hosts to search for.

Click **Remove -** to remove a selected host from the list.

Restrict connections to the following IP addresses: When this PC is a host, you can restrict access to the host by typing the pcAnywhere remote name, IP address, or subnet address of the pcAnywhere remotes that are allowed to connect.

 By creating a list of allowable remotes, you prevent remotes not on the list from connecting to the host PC.

Terminal Emulation Tab

Choose options affecting display and functions during sessions with online services.

Some of these options can be changed during a session by choosing Display from the Session menu of the Terminal window.

- **Automatic Font Sizing**: When checked, pcAnywhere automatically selects the font size displayed on the remote screen, based upon the size of the Terminal window.
- **Automatic Scroll Bars**: When checked, screen displays horizontal and vertical scroll bars.
- **Status Line**: When checked, the online status bar displays at bottom of screen.
- **Macro Keys:** Click to choose a file and specify macro keys during terminal emulation.
- **Translation:** Click to choose a file and specify codes in a translation table.
- **Font/Colors:** Click to choose fonts and color of text during terminal emulation.

File Transfer

Choose options affecting file transfer in all remote control sessions and sessions with online services.

Some of these options can be changed during a remote control session using the remote online toolbar.

Pause Remote Control to Maximize File Transfer Performance: Check this option to optimize file transfer performance. If file transfer speed is not a priority and you want to run simultaneous remote control and file transfer, leave this option unchecked.

Virus Check All Downloaded Files: Check to virus check all downloaded files automatically.

Default Download Folder: Choose a folder on your PC where files downloaded from online services are to reside. Click Browse to find a folder.

Protocols:

To Customize Transfer Protocols:

1 Select a protocol.

2 Click Settings.

3 Set options for the selected protocol.

- Only ASCII, pcAnywhere, and ZMODEM can be customized.
- Customize the pcAnywhere protocol to use SpeedSend
- .

ASCII Protocol Settings dialog box

- Pacing specifies a delay after each line or character that is sent.
- **Character:** Enter the number of tenths of seconds to pause after a character is sent.
- **Line:** Enter the number of tenths of seconds to pause after a line is sent.
- **Pace Character:** Enter the ASCII value of the character that causes transmission to pause until the character is received.

Download timeout: Enter the maximum number of seconds to allow for inactivity before transfer is terminated.

Upload and Download Translations compensate for differences in how carriage returns and linefeeds are handled on different systems. When uploading files (transferring files to the online service), specify how to treat these characters.

- **CR:** Select the way to handle a carriage return.
- **None:** Transmits the character as itself.
- **Strip:** Removes the character from the transmission.
- **Add LF:** Adds a linefeed to the character as it is transmitted.
- **LF:** Select the way to handle a linefeed. Select None, Strip or Add CR.

Drive Access dialog box

Allow this caller any of the following drive access rights:

- **No access**: denies the remote caller from accessing the selected drive.
- **Read only**: allows the remote caller read only access to the selected drive.
- **Full access**: allows the remote caller full access to the selected drive.

COM Port Details

To customize this port for connections using a cable:

Set these options:

- **Speed** - This is the data rate for sessions when this port is used to make a connection.
- **Parity** - Set to same parity as that of the system to which you are connecting. When making a connection with a pcAnywhere host or remote PC, set to None.
- **Flow control** - Transmissions over data rates of 9600 baud usually require that you set flow control to RTS/CTS.
- **Started by** - Do not set for cable connections.
- **Ended by** - Do not set for cable connections.

To customize this port for connections using a modem:

You can do this **only** if the Use modem checkbox appears on this tab. If it does not, close this dialog box and add your modem to your Windows configuration through Modems in the Control Panel. Then choose your modem from the device list in the Connection Info tab.

1 At the bottom of the Details tab, check Use modem.

2 Choose your modem from the drop-down list below the Use modem checkbox. (If your modem is not on the list, check with your modem's manufacturer to find out what modem's configuration most nearly matches yours, or try Hayes Compatible.)

3 Set these options:

Speed - This is the data rate for sessions when this port is used to make a connection. It was selected automatically for you when you chose your modem.
Parity - Set to same parity as that of the system to which you are connecting. When making a connection with a pcAnywhere host or remote PC, set to None.
Flow control - Transmissions over data rates of 9600 baud usually require that you set flow control to RTS/CTS.
Started by - Selected automatically for you when you chose your modem.
Ended by - Selected automatically for you when you chose your modem.

4 Optional: Click Advanced to set more options for your modem.

ISDN via CAPI Details tab

- **Attempt channel bonding:** Check if you want pcAnywhere to attempt to make a 128K connection by using two 64K channels. Channel bonding does not occur if the second channel is unavailable at the time of the connection. Leave this option unchecked if you want the second channel to remain free for other communications.

Check the transmission rate to determine if channel bonding was successful.

- **Only receive calls on extensions:** Type in Multiple Subscriber Number (MSN) extensions, separated with a comma, if you want to restrict incoming calls to only those extensions. Leave this field blank to connect on any available extension.

Network - Details tab

Set options on this tab only if the device you are customizing must use a gateway to make connections.

- **Use gateway checkbox:** Check to specify that the device you are customizing makes connections using a pcAnywhere gateway.
- **Gateway:** Enter information below to identify which gateway, or group of gateways, this PC should look for when making a connection.
- **Name:** Type the name of the gateway. This is the name of the gateway connection item located on the gateway PC.
- **Class:** Type class of gateway this PC must use. If more than one gateway is active on your network, you can use any of them in this class to make a connection. (Since gateways can only support one connection at a time, specifying a class can help you make a connection with whatever gateway is available for use.)

  A gateway's class (if there is one) is also contained in the gateway connection item located on the gateway PC.
- **Parity**glossary_parity: Specifies what a gateway's parity must be for this PC to use it when making a connection.

NetBIOS Options dialog box

Use this dialog box to use a LAN Adapter Number (LANA) that is configured in the Windows 95 or Windows NT operating system.

Selection dialog box

Choose an available ACS component from those listed here.

pcAnywhere Protocol Settings dialog box

Use this dialog box to control how aspects of file transfer should be handled.

- **Destination File:** Select the method by which pcAnywhere decides to overwrite duplicate files in the destination folder. Normally you should require verification before overwriting files. If you intend to overwrite a large number of files you may want to overwrite automatically or overwrite older files only.

Never Overwrite Duplicate Files: Does not overwrite duplicate files in the destination folder.
Always Overwrite Duplicate Files: Automatically overwrites duplicate files in the destination folder.
Verify Before Overwriting: Asks you for confirmation before overwriting a duplicate file.
Always Ask for Destination: Asks you where to place the file.
Overwrite Older Files Only: Automatically overwrites duplicate files that are older than the file you are transferring.

- **Use Compression:** Check to compress files during transfer. Files are compressed during transfer to improve speed, but are automatically uncompressed to their original state before being written to the destination drive.
- **Allow Crash Recovery:** Check to allow reconnection and continued file transfer from the point at which file transfer was interrupted.
- **Use SpeedSend:** Check to use pcAnywhere's SpeedSend

to accelerate transfer to a location where earlier versions of the same files have been sent previously.

Translation Table dialog box

Use this dialog box to define a translation table to translate how your PC sends and receives codes during sessions with online services.

ZMODEM Protocol Settings dialog box

Settings

- **Error Checking:** Select **32-bit** or **16-bit**. 32-bit error checking offers the highest level of error correction.
- **Data Window:** Select **None**, **2K** or **4K**. Use of a Data Window increases reliability, by pausing for an acknowledgment from the receiving computer after a specified amount of data has been sent. Select None to halt transmission only when a negative acknowledgment (NAK) is received by the sending computer.
- **Crash Recovery:** Allows interrupted file transfers to be restarted without data loss.
- **Allow Auto Downloading:** Allows the online service to send files selected for transfer without remote user initiation of the Receive File command.

DOS Sessions

Select these options for remote control sessions during which you use DOS applications:

- **Special keyboard handler**: Some applications, such as terminal emulation programs, handle keyboard input in a non-standard fashion and require a keyboard handler to allow the remote user to effectively run DOS applications on the host PC. In pcAnywhere, the Level 1 keyboard handler works well with most applications.
- **Synchronize display with host**: Use this option to slow the host application to the rate at which the remote can display screen activity. If unchecked, some lines of text may scroll off the screen before the remote is able to display them.
- **Full graphics support**: Select when you want the most accurate representation of DOS graphics possible. Leave unchecked to improve screen display speed if an approximation of graphics is acceptable.

Macro Keys dialog box

Use this dialog box to automate sessions with online services by sending text, executing programs, or running a script using keystrokes you pre-define here.

Activity Log Processing

Use the Activity Log Processing dialog box to create a report from a session log and to delete or archive log data. Using pcAnywhere's log processing, you can track user access for billing or security purposes. After the report is created, you can view it or print it.

To process a session log:

- Choose one:
- **Report**: Displays the Select Destination File for Output dialog box which is a standard browse dialog box. Enter .LOG filename to create a new report. You then have the option to view and then print the report.
- **Archive/Delete**: Displays the Archive/Delete Host/Remote Log dialog box when the log to process is the remote session log or host session log. Use it to archive or delete data from a remote or host session log.

Archive/Delete Host/Remote

Use this dialog box to move older log information (specified by date) into an archive file. The information is preserved in case you need to reference it at a later date. You can also delete information instead of archiving it.

Date Range

- **Start**: Enter the starting date: year, month and day.
- **End**: Enter the ending date: year, month and day.

Delete Log Entries from source: Check to remove the original log entries from the log file and reduces the size of the log file.

Copy Log Entries to archive file: Check to store old log entries in a separate compressed archive file.

Session Log Report

Set options here to format your session log report for sessions either as a host or a remote Pc.

To format your session log report:

1 Choose a report type from the available options:

- **Fully Formatted List** - formats data in a report format, with a summary of events.
- **Data Only - Comma Delimited** - formats data for use in database applications or other programs requiring delimiters between fields. Each item is enclosed in double quotes, followed by a comma. Month, date, and year are separated by slashes (/). Each entry is followed by a carriage return.
- **Data Only - Fixed Fields** - formats each item in a field of fixed size.

2 Set the data range to include sessions from the time period you want:

3 Check options to include the fields you want in either the Remote Options or the Host Options column and click OK.

You are prompted to choose whether to view the report at this time. If you do, the Script Editor window opens and displays the report.

Playback Options

Set options here for the playback of a session you have recorded:

1 Check **Control Panel** to display the Playback Control Panel.

2 Choose which session in the record file you want to begin viewing.

3 Choose whether to play the session over and over, to play all sessions in the record file over and over, or to stop after each session.

4 Choose the speed of the playback. (You can also change this using the Playback Control Panel during the playback.)

AutoTransfer contents tab

The contents tab displays the location and filename of the selected AutoTransfer procedure and lists the file transfer commands in the procedure.

- Click Add - to add commands to the procedure.
- Click Remove - to delete selected commands from the procedure.
- Click Modify - to edit a selected command in the procedure.
- Click Move Up or Move Down to move a selected command to the order in which you want it executed.

File transfer commands are run in the order they are listed, from top to bottom.

AutoTransfer options tab

File transfer options can be changed for this AutoTransfer procedure.

- Click **Use application defaults** to use the default settings in the Application Options, File Transfer property page:
- Click **Override file transfer options for this Automated Transfer** to change the default settings for this AutoTransfer procedure:
- **Destination File -** Select the file overwrite option to use when a duplicate filename is found.

If you want this procedure to run unattended, it is important that you select an automatic file overwrite option. If you use the default setting of Verify before overwriting, the AutoTransfer procedure will stop and wait for confirmation before continuing..

- Check any of the following options:

  Use Compression — to improve file transfer speed by compressing files during the transfer. Files are uncompressed to their original state before being written to the destination folder.

  Allow Crash Recovery — to have pcAnywhere continue transferring an interrupted file transfer at the point of the interruption. If crash recovery is not enabled, an interrupted file transfer starts at the beginning of the file. You should consider using this option when transferring very large files.

  Use SpeedSend — to have pcAnywhere compare duplicate filenames and transfer only the data that is different in the source file.

  Virus Check All Downloaded Files — to allow Norton AntiVirus to scan all files for viruses before downloading.

  End session when finished to automatically disconnect the session when the AutoTransfer procedure is completed.

When you choose to automatically end the session, pcAnywhere ignores all file transfer errors to prevent an unattended procedure from pausing when errors are encountered. The command generating the error is skipped and the procedure continues with the next command. To verify which commands were transferred, enable the Session Log File option in the Automated Tasks property page.

Script dialog box

From the Scripts dialog box, you can create a new script, or select a script you have already created and edit, compile or run it. Or, you can select an existing script file to run after connecting with a specific host or online service.

This dialog box contains a list box of all currently defined scripts, with a short description of each. Directly beneath the list box is the Script Path. If your script files are located in a folder other than the one displayed, you can change folders using the Browse button.

To perform an action on a script:

1 Choose a file in the list box.

2 Click either Edit, Compile, Run or Select to perform that function on the selected script.

Select ASCII Data File dialog box (Standard Browse)

Use this dialog box to save screen data to an ASCII file.This dialog is called from the playback screens window by choosing Save screen as ASCII from the File menu.]]

- **File Name**: Type or select the filename you want to open. This box lists files with the extension selected in the List Files of Type box.
- **List Files of Type**: Select the type of file you want to open.
- **ASCII Data Files (*.TXT):** Lists all files in the current folder that were saved with a *.TXT extension.
- **Drives**: Select the drive in which pcAnywhere stores the file.
- **Folders**: Select the folder in which pcAnywhere stores the file.

Terminal window

Conduct sessions with online services from this window.

- You can adjust terminal emulation and file transfer protocol settings during your session. (Click the How do I... button above for more information.)

Hardware Settings dialog box

Use the Hardware Settings dialog box to change certain communications parameters while remaining connected to an online service.

Settings group box

- **Data rate** drop-down list box: Select the highest speed (in bps) at which your modem and COM port can transmit and receive data. If you connect to a slower modem, your modem automatically adjusts its speed.
- **Parity:** Select the type of error checking performed during the communications session. This setting must match that of the computer to which you are trying to communicate. You may need to check with the system administrator of the online service to determine the correct setting.

- **None:** No parity error correction is used. Use this method for connecting to most bulletin boards (BBSs). This is the equivalent to 8 data bits, no parity, and 1 stop bit—also known as 8,n,1.
- **Odd:** A parity bit is added to the bits that make up each character to adjust the number of 1's in each set of bits to an odd number. Use this method when required by an online service. This is also known as 7, O, 1.
- **Even:** A parity bit is added to the bits that make up each character to adjust the number of 1's in each set of bits to an even number. Use this method when required by on-line services such as CompuServe. This is also known as 7, E, 1.
- **Mark:** A parity bit is used and is always set to 1. Use this method when required for communications with mainframe or minicomputers.
- **Space:** A parity bit is used and is always set to 0. Use this method when required for communications with mainframe or minicomputers.

- **Flow Control:** Select how to regulate the flow of data between computers. This prevents data loss when data is sent faster than the receiving computer can handle. This is sometimes called handshaking. You may need to check with the system administrator of the online service to determine the correct setting.

- **None:** No flow control is used. Use this if your modem is operating below 9600 bps.
- **XONXOFF:** Uses software to send control codes between machines to pace the flow of data. Also called software flow control. Use this when required by an online service.
- **RTS/CTS:** Uses the RTS and CTS lines of the RS-232 interface to pace the flow of data. Also called hardware flow control. This is the method used by most high-speed and error-correcting modems, and is usually the correct choice for connecting with online services.
- **Both:** Uses both software and hardware flow control. Use this when required by an online service.

Terminal Settings

Use the Terminal Settings dialog box to change the terminal emulation type or re-configure the current terminal emulation type options while remaining connected to an online service.

Emulation: Select the type of terminal emulation as required by the online service to which you are connected.

Choose Font: Displays Font dialog box.

Options

- **Line wrap:** Advances the cursor to the first column of the next line, when it reaches the last screen column. If this option is disabled, the cursor will stay in the last column and each new character will overwrite the previous one.
- **Destructive BS:** Causes the backspace key to delete characters as it moves the cursor to the left. This works like sending a destructive backspace code (127). If disabled, the backspace key will move the cursor to the left without deleting a character. This works like sending a non-destructive backspace code (08).
- **Screen wrap:** Advances the cursor to the first column of the first row when it reaches the last column of the last row. Subsequent text will overwrite whatever appears on the first row. If disabled the top line of the display scrolls upward, out of view when the screen is filled, making room for an additional line of text at the bottom.
- **CR to CR/LF:** Moves the cursor down one line after processing the carriage return code. If disabled, the cursor moves to the first position on the current line.

Advanced Options

- **Macro Keys:** Enables the user-defined macro keys. Macro keys can be defined to transmit a string of commands, start a script or run a program on the online service.
- **Translation Tables:** Uses the translation table specified in the Advanced Settings for Online Service dialog box. Translation tables redefine the hexadecimal codes of incoming or outgoing characters. A translation table allows you to make a terminal emulation match specific online service requirements.

Font dialog box

Use the Font dialog box to change the current terminal emulation font type.

- **Font:** Select the name of a font to be used to display text in the Terminal window.
- **Font Style:** Select the style to be applied to the selected font, such as bold or italic.
- **Size:** Select the size of the type to be used to display text in the Terminal window.
- **Sample:** Displays a type sample using the selected font, style and size.

Display Settings dialog box

Use the Display Settings dialog box to set certain Terminal window display options while connected to an online service.

- **Vertical Scroll Bar:** Enables the vertical scroll bar that is used to view text that has scrolled vertically out of view in the Terminal window.
- **Horizontal Scroll Bar:** Enables the horizontal scroll bar that is used to view text that has scrolled horizontally out of view in the terminal Window.
- **Status Bar:** Enables the Terminal window status bar which indicates the current settings for terminal emulation type, COM port, parity and flow control.

Select Files For Transfer dialog box (Standard Browse)

Use this dialog box to send or receive files from an online service.

Sending Files

- **File Name:** Type or select the name of the file or files to send to the online service. If you have selected YMODEM or ZMODEM as your file transfer protocol, you may select more than one file and transfer them as a group. This is called a batch transfer.
- **List Files of Type:** Select or enter the type of file you want to display in the list box.
- **File Transfer (*.*):** Lists all files in the current folder.
- **Drives:** Select the drive on which the files are stored.
- **Folders:** Select the folder in which the files are stored.

Receiving Files

- **File Name:** The text box displays the name of the file that you have requested from the online service. The list box displays the contents of the folder on your computer to which the file will be saved.
- **List Files of Type:** Select or enter the type of file you want to display in the list box.
- **File Transfer (*.*):** Lists all files in the current folder.
- **Drives:** Select the drive to which you wish to save the file downloaded file.
- **Folders:** Select the folder in which you wish to save the file downloaded file.

File Transfer Protocol dialog box

Protocols

Select a protocol to use for file transfer.

- **XMODEM:** A commonly used file transfer protocol that transfers one file at a time in blocks of 128 bytes. Do not use when transferring files with CompuServe or over asynchronous outdial networks like Tymnet.
- **XMODEM-CRC:** A modified version of the original XMODEM with better error correction. Do not use when transferring files with CompuServe or over asynchronous outdial networks like Tymnet.
- **XMODEM-1K:** A modified version of the original XMODEM that transfers files in 1024 bytes. Do not use when transferring files with CompuServe or over asynchronous outdial networks like Tymnet.
- **XMODEM-1K-G:** A version of XMODEM 1K without error correction. Select only when both you and the Online Service are using error-correcting modems which use MNP or V.42 error correction. Do not use when transferring files with CompuServe or over asynchronous outdial networks like Tymnet.
- **YMODEM (batch):** Features high reliability, 1024 byte transfer packets and multiple file transfers with a single command (batch transfers).
- **YMODEM-G (batch):** Features 1024 byte transfer packets and multiple file transfers with a single command (batch transfers), without error correction. Select only when both you and the Online Service are using error-correcting modems which use MNP or V.42 error correction.
- **ZMODEM:** A very reliable feature packed file transfer protocol. Its extremely robust error correction make it an excellent choice where excessive line noise is a problem. Protocol options may be specified in the Settings Dialog group box.
- **KERMIT:** Originally developed for communication between mainframe computers, this popular public domain protocol has been adapter for use on virtually every type of computer.
- **ASCII:** Refers to the transmission of plain text files using minimal error correction. Protocol options may be specified in the Settings Dialog group box. It is usually preferable to use a binary file transfer protocol (those listed above) if it is available.

Settings (ZMODEM)

Both ZMODEM and ASCII protocol selections have user defined settings.

- **Error Checking:** Select **32-bit** or **16-bit**. 32-bit error checking offers the highest level of error correction.
- **Data Window:** Select **None**, **2K** or **4K**. Use of a Data Window increases reliability, by pausing for an acknowledgment from the receiving computer after a specified amount of data has been sent. Select None to halt transmission only when a negative acknowledgment (NAK) is received by the sending computer.
- **Crash Recovery:** Allows interrupted file transfers to be restarted without data loss.
- **Allow Auto Downloading:** Allows the online service to send files selected for transfer without remote user initiation of the Receive File command.

Settings (ASCII)

Pacing specifies a delay after each line or character that is sent.

- **Character:** Enter the number of tenths of seconds to pause after a character is sent.
- **Line:** Enter the number of tenths of seconds to pause after a line is sent.
- **Pace Character:** Enter the ASCII value of the character that causes transmission to pause until the character is received.

Download: Enter the maximum number of seconds of inactivity before transfer is terminated.

Upload Translations

Upload and Download Translations compensate for differences in how carriage returns and linefeeds are handled on different systems. When uploading files (transferring files to the online service), specify how to treat these characters.

- **CR:** Select the way to handle a carriage return.
- **None:** Transmits the character as itself.
- **Strip:** Removes the character from the transmission.
- **Add LF:** Adds a linefeed to the character as it is transmitted.
- **LF:** Select the way to handle a linefeed. Select None, Strip or Add LF.

Download Translations

When downloading files (transferring files from the online service), specify how to treat these characters.

- **CR:** Select the way to handle a carriage return. Select None, Strip or Add LF.
- **LF:** Select the way to handle a linefeed. Select None, Strip or Add LF.

Administrator Password

This dialog box displays the first time you run pcAnywhere on a workstation where a network installation took place. You must set a password here to access administrative features within the pcAnywhere program.

Any user running pcAnywhere on this machine will be prompted for the password you record now. This password **is** case-sensitive.

To record the administrator password:

1 Type the password you want:

2 Re-type the same password, remembering that administrator passwords **are** case-sensitive:

- If you click OK without entering a password, any user running pcAnywhere can access administrative features on this machine.
- If you assign a password and another user who does not know it runs pcAnywhere on this machine, he or she can click OK to bypass the password prompt and still use the program -- without administrative features.

Network Host mode

1 Check **Allow folder browse**: to allow pcAnywhere users on your network to display and use folder tools on their toolbar when in the host mode.

- If you check this option, users can choose any folder to store their host connection items, and use items stored by other users in folders to which they have access.
- Users may choose to display the folder tools in the Button Bars tab, but if you leave this item unchecked, the tools will not display in the host mode.

2 Choose a default folder to store host connection items for all pcAnywhere users on your network.

Network Remote mode

1 Check **Allow folder browse:** to allow pcAnywhere users on your network to display and use folder tools on their toolbar when in the remote control mode.

- If you check this option, users can choose any folder to store their remote control connection items, and use items stored by other users in folders to which they have access.
- Users may choose to display the folder tools in the Button Bars tab, but if you leave this item unchecked, the tools will not display in the remote control mode.

2 Choose a default folder to store remote control connection items for all pcAnywhere users on your network.

- The options you choose on this tab also affect the use of file transfer connection items for all users.

Network Gateway mode

1 Check **Allow folder browse:** to allow pcAnywhere users on your network to display and use folder tools on their toolbar when in the gateway mode.

- If you check this option, users can choose any folder to store their gateway connection items, and use items stored by other users in folders to which they have access.
- Users may choose to display the folder tools in the Button Bars tab, but if you leave this item unchecked, the tools will not display in the gateway mode.

2 Choose a default folder to store gateway connection items for all pcAnywhere users on your network.

Network Online mode

1 Check **Allow folder browse**: to allow pcAnywhere users on your network to display and use folder tools on their toolbar when in the online service mode.

- If you check this option, users can choose any folder to store their online service connection items, and use items stored by other users in folders to which they have access.
- Users may choose to display the folder tools in the Button Bars tab, but if you leave this item unchecked, the tools will not display in the online service mode.

2 Choose a default folder to store online service connection items for all pcAnywhere users on your network.

Computer Name Selection

Choose this option:

to allow any network user of pcAnywhere to change the name identifying his or her PC in remote control sessions. (The user sets this name in the System Setup tab of the Application Options dialog box.)

List of network hosts

You are using a network connection device to establish a remote control session.

 Click an available host or gateway, then click OK.

 A host or gateway displaying with this symbol is not available at this time, although it is active on the network.

 A host that is in conference with one or more remote callers.

 A host or gateway displaying with this symbol may be available to you.

Host online menu

Choose one of the following options from the host online menu:

- **End session -** To disconnect from the remote PC.
- **File Transfer** - To open the File Manager window on either the host or remote in order to send files from one PC to the other.
- **Chat** - To open a window to type a conversation with a user at the remote PC.

Chat window

To chat with a user at the other PC:

1. Type your comments to the user at the PC to which you are connected.
2. Press Enter to send your comments to that user.
3. When you are finished chatting, close the Chat window.

- Closing the window on the remote PC automatically closes it on the host PC.

Dial-Up Networking

Use the settings on this property page to allow a pcAnywhere remote to connect to a pcAnywhere host on a network or on the Internet.

- **Connect to a pcAnywhere host using a local area network:** Select this option to make network connections to a server or ISP (Internet Service Provider).
- **Connect to a pcAnywhere host using a modem:** Select this option to make a modem connection to a server or ISP (Internet Service Provider).
- **Use the following Dial-Up network connection**: Choose the dial-up networking configuration to use to make a connection to a network server or Internet Service Provider (ISP).

  [optional] Click Add to configure a new dial-up network configuration.

  [optional] Click Properties to view the dialing properties of the selected configuration.
- **Number of times to attempt connection**: Set the number of time the remote redials to try and connect to the host
- **Number of seconds to wait between attempts**: Set the number of seconds to wait between redial attempts.
- **Log on information**: Enter the User name, password, and domain [optional], used to connect to the host.
- **Disconnect if idle for**: Set the number of minutes to allow for inactivity before disconnecting the session.

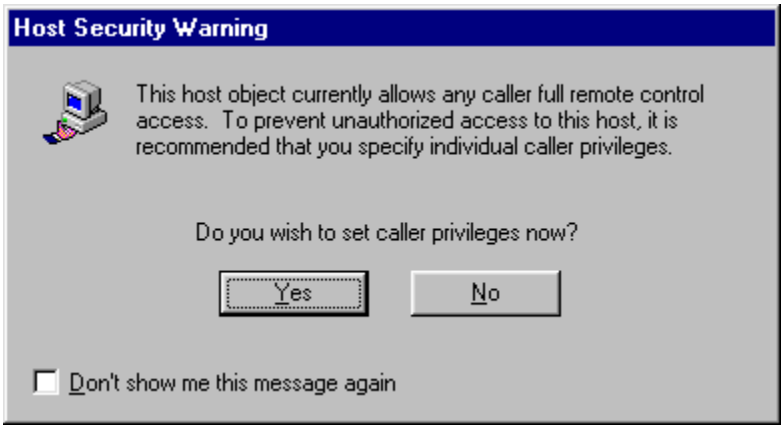
Host Operations

Ring number to answer on: set the number of rings to allow before the host answers a call.

Video mode selection:

- Default (Accelerator enabled): Maximizes the performance of a pcAnywhere remote control connection. This is the preferred video mode and should be selected unless you experience video problems on the host PC.
- Accelerator disabled: Select this option if the host is using a 3D accelerator video card. NOTE: This option is only available for Windows 9x hosts.
- Compatibility: Select this option if you experience incomplete or distorted video display when connected to the host PC.

Show warning dialog when caller security is not enabled: Check this option to display the following warning message whenever an unsecure host is launched.



Remote Printing

Remote printing allows the remote caller to print to the local printer during a session. Select the printers on the remote PC you want to print to during a remote control session.

- The remote's printer definitions must first be installed on the host PC using the Windows 95 or Windows NT Control Panel.

1 Click Add to add a printer definition.

2 Do one of the following:

- Select the remote printer driver from the list of Currently Installed Drivers and click Next.
- Select the manufacturer and model of the remote printer from the Manufacturers and Printers list boxes and click Next.

3 Type a descriptive name that identifies the owner or location of the remote printer and click Finish. The remote caller selects this name to print to the local printer during a session.

Host Conference tab

A conference host allows multiple remote users to simultaneously connect and view the host's activities. A conference host is remotely controlled by the first remote caller that connects. Other callers can connect and view the host session but they cannot control the actions on the host PC.

- To enable conferencing for this host PC check **Enable conferencing**.
- To have the network server automatically assign an available multicast IP address click **Obtain IP address automatically**.
- To use a specific multicast IP address click **Specify IP address** and type the the address in the text box. Multicast IP addresses must be within the range of 225.1.1.1 and 239.254.254.254
- To specify the number of routers and increase the broadcast area of this conference, enter the number allowed in the **Allow conference over routers** box.
- The first caller can use any connection device to connect to the conference host. All other callers must use TCP/IP network connections to the host.

DOS Sessions

When the host is running DOS in a full-screen window, or when the host user has exited to DOS, the remote may experience problems reading and displaying the host screen. The following options allow the remote user to make adjustments that allow correct viewing of these DOS display characters:

- **Special keyboard handler**: Some applications, such as terminal emulation programs, handle keyboard input in a non-standard fashion and require a keyboard handler to allow the remote user to effectively run DOS applications on the host PC. In pcAnywhere, the Level 1 keyboard handler works well with most applications.
- **Synchronize display with host**: Use this option to slow the host application to the rate at which the remote can display screen activity. If unchecked, some lines of text may scroll off the screen before the remote is able to display them.
- **Full graphics support**: Select when you want the most accurate representation of DOS graphics possible. Leave unchecked to improve screen display speed if an approximation of graphics is acceptable.

Font/Colors: Use to access the Fonts dialog box and set colors, and fonts for remote control sessions of hosts running DOS programs.

Remote Security Options tab

Set the level of encryption this connection item uses for every session.

Encryption Level

Public Key encryption: Provides the highest level of security and is used when a certificate authority makes public keys available to the Cryptographic Service Provider (CSP) on both host and remote sides of the session.

Symmetric encryption: Provides the next level of security and is used when there is no certificate authority available but there is a CSP.

PcAnywhere encryption: Provides minimum encryption capability and is used when there is no CSP available. It is the only level compatible with pcAnywhere versions 2.0, 5.0, and 7.x.

Deny Lower Encryption Level: Check to prevent connections with PCs using a lower level of encryption.

Certificate Common Name: Type the name provided to the certificate authority at the time the certificate was requested. This option is case sensitive, type the name exactly as you provided it to the certificate authority.

NT Caller Identification Tab

The Identification property page displays the configuration used for the selected caller and includes the caller's domain name NT User Account name.

- This property page is for information only and cannot be edited. To change the configuration for this caller, click the Add User wizard to create a new caller item.

General tab

You can temporarily modify the following settings for the current session only:

- Check **Screen scaling** to reduce the host screen to fit in the remote's window.
- Check **Remote control toolbar** to display the remote control toolbar on the remote PC.
- Check **Remote control status tray** to display the remote control status tray on the remote PC.
- To control the performance of a session, choose the number of host colors to display from the drop-down menu. Using fewer colors improves the session performance:
- Check **Reduce host desktop** to match remote to reduce the host screen resolution to match the resolution used on the remote PC.
- Check **Host active window tracking** to automatically move the remote's view to any currently active windows on the host PC. For example, a remote caller could be unaware of an error message if it is displayed beyond the remote's current viewing area. If this option is checked, the remote's viewing area is automatically focused on the error message.
- Check **Host keyboard locked** to prevent a user at the host PC from using the keyboard on the host.
- Check **Host screen blanked** to blank the host's screen during a session, to prevent user at the host site from viewing session activities

Terminal Emulation Tab

Application Options / Host Logging Options

Host Session Recording

A Host Session recording file allows you to create a video recording of all activities occurring during a session with a host PC. This file can be played back after the session ends and used to troubleshoot a complex program or procedure run on the host PC.

1 Check **Record host session in files for later playback**: to start recording sessions after connecting to the host PC.

2 Type the folder where you want to store the pcAnywhere recording file.

- Click Advanced to type the name, password, and domain information required to access this network destination.

PCA Logging

The pcAnywhere log file allows you to capture information about remote control sessions for historical or security purposes. Information in the log file is viewed by generating an activity log report.

1 Check Enable pcAnywhere Log Generation to capture the selected events to a pcAnywhere log file.

2 Do one of the following:

- Check the pcAnywhere events you want to logged to the pcAnywhere log file.
- Click Select All to select all events.

3 Check **pcA log locally**: to create the pcAnywhere log files on the local computer.

4 Check **Create pcA log on central server**: to save the pcAnywhere log files to a folder on the network.

- Click Advanced to type the name, password, and domain information required to access this network destination.

SNMP Logging

SNMP (Simple Network Management Protocol) is a Windows component allowing a computer to be monitored remotely with third party management tools.

1 Check **Enable SNMP traps** to specify the pcAnywhere traps to generate in a log file.

- **Note**: A central management utility such as Microsoft Management Console (MMC), Microsoft Systems Management System (SMS), Computer Associates UniCenter TNG, or IBM Tivoli must be installed to use SNMP logging.

2 Do one of the following:

- Check to select the pcAnywhere traps you want logged.
- Click **Select All** to check all traps.
- Click Deselect All to uncheck all selected traps.
- Click to check all unselected traps and uncheck all selected traps.

3 Click Add to specify the computer IP address for where you want trap information sent.

NT Event Log

1 Check **Enable NT Event Logging** to send pcAnywhere events to the Windows NT Event log.

2 Do one:

- Check the pcAnywhere events you want to log.
- Click **Select All** to select all events.

Directory Services

Note: Before you can configure pcAnywhere to use directory services, the directory server entries must be configured by a network administrator.
Do any of the following:

- Click Remove to remove a directory server from the list.
- Click Properties to view the configuration of a directory server on the list.
- Click Add to add a directory server to the list and type the following information for the new server:

  Display name: Type a descriptive name to identify the directory server. This name appears in the Use directory services list box in the Settings property page of a remote control connection item.

  Directory server: Type the host name or IP address of the directory server.

  Authentication Name & Password: Type the name and password used when the remote caller logs on to the LDAP server.

  Click Advanced to set:

  Port Number: The port that the directory server uses to accept queries from the client. The default port is 389.

  Search base: Type the root of the directory structure where search queries begin.

Save Recorded Script

Use this dialog box to:

- **Save Script As** : Saves the recorded script to a file for later playback.
- **Discard Script**: Deletes the recorded script.
- **Continue Recording**: Continues recording the current session.

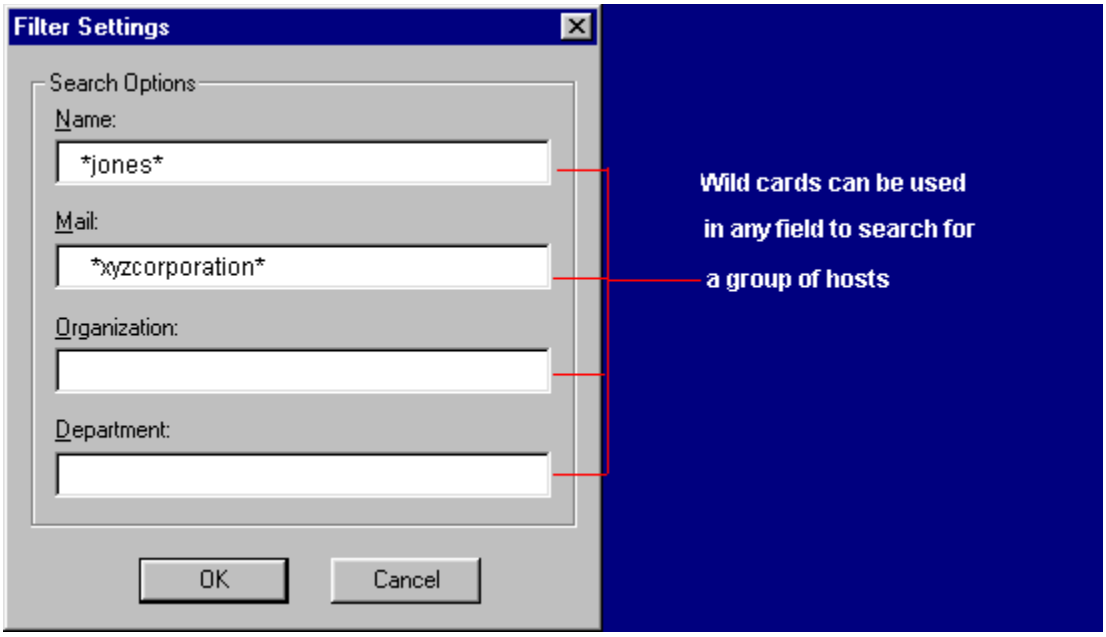
Directory Server Browser

A list of pcAnywhere hosts available on the selected directory server.

- Click an available host or gateway, then click OK.
- A host or gateway displaying with this symbol is not available at this time, although it is active on the network.
- A host that is in conference with one or more remote callers.
- A host or gateway displaying with this symbol may be available to you.

To search for hosts on the directory server:

1 Click Search for.



2 Search for a pcAnywhere host using any of the following information:

- **Name**: Type a user name to search for or use wild cards to display a group of users. For example: ***jones*** lists all users with the jones string in its content.
- **Mail:** Type the user's email address or use wild cards to display a group of email addresses. For example: ***xyzcorporation*** lists all email addresses containing the xyzcorporation string.
- **Organization**: Type the host's organization name or use wild cards to search for a group of users within a specific organization.
- **Department:.**Type a user's department or use wild cards to search for a group of users within a department.To connect to a pcAnywhere host:

3 Click the host to select it.

4 Click OK.

Playback Control Panel

Introducing pcAnywhere

pcAnywhere offers a total communications solution for your remote control, file transfer, and remote networking needs.

With pcAnywhere you can:

- Remotely control one PC from another, as long as both PCs are running pcAnywhere software.
- Automatically transfer files and synchronize folders between two PCs.
- Connect to a variety of online services.
- Become a workstation on a network using a remote networking connection.

You can make connections using telephone lines, networks, or a combination of the two. You can also make direct connections between two PCs using either a serial or parallel cable. For example, you may want to transfer files between your laptop and desktop PC.

The pcAnywhere Host PC

The pcAnywhere host is waiting for a remote caller to login with the user name and password specified in the host Callers property page.

The pcAnywhere Host PC

The pcAnywhere host is currently in session with a remote caller. As the host user you can:

- Select **End Session** to disconnect the current session and wait for another call.
- Select **File Transfer** to initiate a file transfer and choose whether the host or remote user controls the file transfer actions.
- Select **Chat** to open a chat window and have a typed conversation with the remote caller.

The pcAnywhere Host PC

The pcAnywhere host is currently waiting for a connection. Click Display Status to view the host name and connection device selected for the connection.

Script Editor Commands Listed Alphabetically **{button Click to switch to task list.,JI(`',`commands_by_task')}**

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

-A-

Answer

-B-

Beep

Break

-C-

CD Cursor Down

Clear BOL

Clear BOP

Clear EOL

Clear EOP

Clear Screen

Close

Copy

Create

Cursor block

Cursor Home

Cursor Left

Cursor Line

Cursor Off

Cursor Position

Cursor Restore

Cursor Right

Cursor Save

Cursor Up

-D-

Decrypt

Del Dial Number

Delete char

Delete line

Description

Dial Host

Dial OnlSvc

Dir

-E-

Emulate

Encrypt

End

End Menu

End Terminal

Exit

-F-

Find First

-S-

-T-

Terminal
Trim
Type File
Type Line
Type String

-U-

Upper

-W-

Wait Carrier
Wait Receive
Wait Silence
Wait String
Wait Time
Wait Until
Write Line
Write String

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Answer

Description: This command works for all device types. Use it to wait for an incoming call. The command terminates when a connection is made or when the user presses Esc.

The hardware configuration specifies how to detect a successful answer. When creating the hardware configuration in Norton pcAnywhere for DOS, you set the Connection Started By parameter in the Configuration form. In pcAnywhere for Windows, you select an option from the Started By drop-down list box in the Advanced Hardware Options dialog box.

Syntax: **Answer**

Reserved Variables: $Result Set to 0. If an error occurs, set to the value of $Error.

$Error Standard errors. See Appendix B, “Error Messages.”

Example: The following example loads a hardware configuration and waits for a call.

```
Load OnlSvcInfo "ACE BBS”
Answer
```

See Also: Break, Dial Host, Dial Number, Dial OnlSvc, Hang

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Beep

Description:	Use this command to sound an audible alarm on the remote PC. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.
Syntax:	**Beep**
Reserved Variables:	$Result Set to 0.
Example:	The following example beeps twice. Type String "Attention! An error has occurred." Beep Beep
See Also:	Message Box, Type Line, Type String

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Break

Description: This command applies to serial devices only. Use it to place a break signal on the communication line. The duration of the break signal is specified by the $BrkLen reserved variable, and is measured in tenths of a second. By default, $BrkLen is set to 0.

Syntax: **Break**

Reserved Variables: $Result Set to 0.

Example: The following example causes a .2-second break.

```
$BrkLen = 2

Break          ;breaks the connection
```

See Also: Answer, Dial Host, Dial Number, Dial OnlSvc, Hang

Script Editor Commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

CD

Description: This command applies to the remote PC only. Use it to change the current directory on the current drive to the specified directory. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **CD** pathname

Parameters:

pathname	String literal or string variable containing the full pathname of the new current directory. You must include a backslash (\) at the beginning of the pathname for the root directory on the current drive and between directory names. The drive letter is optional.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."
$CurDir	Set to string containing the name of the current directory on the current drive.

Example: The following example changes the current directory on the current drive so you can access files in that directory.

```
CD "\AW"
```

The next example prompts the user for the new current directory.

```
String newpath[66]
;prompt the user for the path
Type Line "Enter a directory: "
;read the pathname into the newpath variable
Input newpath 66
;change the directory using the newpath variable
CD newpath
```

The last example reads the new current directory from a file.

```
String newpath[66]
;open a data file, read the path from it,
;and close the file
Open 1 "datafile.txt"
Read String 1 newpath 66
Close 1
;change the directory using the newpath variable
CD newpath
```

See Also: Dir , MD, RD, SessOpr Host Run, SessOpr Remote Run

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Clear BOL

Description: This command manipulates characters in the remote PC's display buffer. Use it to clear all characters in the current row from the beginning of the row (column 1) up to but not including the current cursor position. Each character in the row is replaced by a normal video space. The cursor position does not change.

Syntax: **Clear BOL**

Reserved Variables: $Result Set to 0.

Example: The following example positions the cursor at row 10, column 15 and uses Clear BOL to clear the characters in row 10, columns 1 through 14.

Cursor Position 10 15

Clear BOL

See Also: Clear BOP, Clear EOL, Clear EOP, Clear Screen

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Clear BOP

Description: This command manipulates characters in the remote PC's display buffer. Use it to clear all characters from the upper-left corner of the screen (row 1, column 1) up to but not including the current cursor position. Each character is replaced by a normal video space. The cursor position does not change.

Syntax: **Clear BOP**

Reserved Variables: $Result Set to 0.

Example: The following example positions the cursor at row 10, column 15 and uses Clear BOP to clear the screen from row 1, column 1 to row 10, column 14.

Cursor Position 10 15

Clear BOP

See Also: Clear BOL, Clear EOL, Clear EOP, Clear Screen

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Clear EOL

Description:	This command manipulates characters in the remote PC's display buffer. Use it to clear all characters in the current row from the current cursor position to the end of the row (inclusive). Each character is replaced by a normal video space. The cursor position does not change.
Syntax:	**Clear EOL**
Reserved Variables:	$Result Set to 0.
Example:	The following example positions the cursor at row 10, column 15 and uses Clear EOL to clear the characters in row 10, starting with column 15 and ending with the last character in the row. Cursor Position 10 15 Clear EOL
See Also:	Clear BOL, Clear BOP, Clear EOP, Clear Screen

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Clear EOP

Description: This command manipulates characters in the remote PC's display buffer. Use it to clear the screen from the current cursor position to the lower-right corner of the screen (inclusive). Each character is replaced by a normal video space. The cursor position does not change.

Syntax: **Clear EOP**

Reserved Variables: $Result Set to 0.

Example: The following example positions the cursor at row 10, column 15 and uses Clear EOP to clear the screen from row 10, column 15 to last column of the last row.

```
Cursor Position 10 15
Clear EOP
```

See Also: Clear BOL, Clear BOP, Clear EOL, Clear Screen

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Clear Screen

Description:	This command manipulates characters in the remote PC's display buffer. Use it to clear the entire screen and move the cursor to the upper-left corner (row 1, column 1). Each character is replaced by a normal video space.
Syntax:	**Clear Screen**
Reserved Variables:	$Result Set to 0.
Example:	The following example clears the screen and places the cursor at row 1, column 1. Clear Screen
See Also:	Clear BOL, Clear BOP, Clear EOL, Clear EOP

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Close

Description: This command applies to the remote PC only. Use it to close the file associated with the specified file number. Once a file is closed, it cannot be read from or written to. Any file left open when script processing terminates is closed automatically. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

TIP: Although it is not necessary to close files when you have finished working with them, it is generally a good practice to do so. When files are open, they are vulnerable to corruption. If your script should terminate unexpectedly, any open files could be damaged.

Syntax: **Close** file_number

Parameters:

file_number	Integer literal used by Open or Create command to identify the file. Valid values are 1 through 9.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, “Error Messages.”

Example: The following example opens TESTFILE.DAT, gives it the file number 1, writes one line, then closes the file.

```
Open 1 "testfile.dat"
Write Line 1 "Sample entry."
Close 1
```

See Also: Copy, Create, Del, Open, Ren

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Copy

Description: This command applies to source and destination files on the remote PC only. Use it to make an exact copy of a DOS file with a different name or with the same name in a different directory. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **COPY** source destination

Parameters:

source	String literal or string variable containing the pathname (optional) and filename of the source file to copy. If only a filename is specified, the file must be in the current directory.
destination	String literal or string variable containing the pathname (optional) and filename for the destination file. If only a filename is specified, the copy is made in the current directory.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example makes a backup copy of REFERENC.TXT.

```
COPY "\docs\referenc.txt" "\olddocs\referenc.txt"
```

The next example prompts the user for the new filename then creates a copy using that name.

```
String newname[12]
;prompt the user for filename
Type Line "Enter the name of the new file: "
;read the destination (from screen)
Input newname
COPY "file1.txt" newname
```

See Also: Close, Create, Del, Open, Ren, SessOpr Host Run, SessOpr Remote Run

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Create

Description: This command applies to the remote PC only. Use it to create and open a new file. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

TIP: Using the Create command on an existing file with read/write attributes is an easy way to clear the file's contents. The existing file length is truncated to 0. If the truncation is successful, the file is opened.

Syntax: **Create** file_number filename [attributes]

Parameters:

file_number	Integer literal used to identify the file. Valid identifiers are 1 through 9.	
filename	String literal or string variable containing the name of the file.	
attributes	(Optional) string literal or string variable containing attributes you want the file to have. The default is non-sharable read/write.	

Use a combination of the following attributes:

R	read-only
W	read/write
S	sharable

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.	
$Error	Set to -9 if the file already exists, except in cases where the attributes variable is set to read/write.	

All other standard errors apply. See Appendix B, "Error Messages."

Example: The following example creates NEWFILE.TXT, a sharable read/write file named in the current directory.

```
Create 1 "newfile.txt" "SW"
```

The next example uses a variable to provide the name of the new file.

```
String filename[12]
;assign the filename to the string variable
filename = "newfile.txt"
;create and open the new file with the
;default setting
Create 1 filename
```

See Also: Close, Copy, Del, Open, Ren, Set File Attr, Set File Date, Set File Size, Set File Time

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Cursor Block

Description: This command applies to the remote PC's display buffer. Use it to change the shape of the cursor to a rectangular block, or to show a cursor hidden using the Cursor Off command.

Syntax: **Cursor Block**

Reserved Variables: $Result Set to 0.

Example: The following example changes the cursor's shape to indicate that the user has entered insert mode.

Cursor Block

See Also: Cursor Down, Cursor Home, Cursor Left, Cursor Line, Cursor Off, Cursor Position, Cursor Restore, Cursor Right, Cursor Save, Cursor Up

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Cursor Down

Description:	This command manipulates characters in the remote PC's display buffer. Use it to move the cursor down the specified number of rows, or one row if no number is specified. If the number exceeds the number of rows below the current position, the cursor moves to the bottom of the screen. The cursor remains in the same column.	
Syntax:	**Cursor Down** [count]	
Parameters:	count	(Optional) integer literal or integer variable containing number of rows to move cursor. The default is 1.
Reserved Variables:	$Result	Set to 0.

Example: The following example moves the cursor down five rows.

```
Cursor Down 5
```

The next example moves the cursor down the number of rows specified in the count variable.

```
Integer count
count = 5
Cursor Down count
```

See Also: Cursor Block, Cursor Home, Cursor Left, Cursor Line, Cursor Off, Cursor Position, Cursor Restore, Cursor Right, Cursor Save, Cursor Up

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Cursor Home

Description:	This command manipulates characters in the remote PC's display buffer. Use it to move the cursor to the upper-left corner of the screen (row 1, column 1).
Syntax:	**Cursor Home**
Reserved Variables:	$Result Set to 0.
Example:	The following example moves the cursor to row 1, column 1. Cursor Home
See Also:	Cursor Block, Cursor Down, Cursor Left, Cursor Line, Cursor Off, Cursor Position, Cursor Restore, Cursor Right, Cursor Save, Cursor Up

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Cursor Left

Description:	This command manipulates characters in the remote PC's display buffer. Use it to move the cursor left the specified number of columns, or left one column if no number is specified. If the number exceeds the number of columns to the left of the current position, the cursor moves to the leftmost column. The cursor remains in the same row.	
Syntax:	**Cursor Left** [count]	
Parameters:	count	(Optional) integer literal or integer variable containing the number of columns to move the cursor. The default is 1.
Reserved Variables:	$Result	Set to 0.

Example: The following example moves the cursor left 5 columns.

```
Cursor Left 5
```

The next example moves the cursor left the number of columns specified in the count variable.

```
Integer count

count = 5

Cursor Left count
```

See Also: Cursor Block, Cursor Down, Cursor Home, Cursor Line, Cursor Off, Cursor Position, Cursor Restore, Cursor Right, Cursor Save, Cursor Up

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Cursor Line

Description:	This command applies to the remote PC's display buffer. Use it to change the cursor shape to a single underscore (underline), or to show a cursor previously hidden by a Cursor Off command.
Syntax:	**Cursor Line**
Reserved Variables:	$Result Set to 0.
Example:	The following example changes the cursor to indicate that the user has entered overwrite mode. Cursor Line
See Also:	Cursor Block, Cursor Down, Cursor Home, Cursor Left, Cursor Off, Cursor Position, Cursor Restore, Cursor Right, Cursor Save, Cursor Up

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Cursor Off

Description:	This command applies to the remote PC's display buffer. Use it to hide the cursor. The cursor position does not change. Show the cursor by using either the Cursor Block or Cursor Line command.
Syntax:	**Cursor Off**
Reserved Variables:	$Result Set to 0.
Example:	The following example hides the cursor. Cursor Off
See Also:	Cursor Block, Cursor Down, Cursor Home, Cursor Left, Cursor Line, Cursor Position, Cursor Restore, Cursor Right, Cursor Save, Cursor Up

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Cursor Position

Description: This command manipulates characters in the remote PC's display buffer. Use it to move the cursor to a specific position on the screen. For example, row 1, column 1 is the upper-left corner of the screen.

Syntax: **Cursor Position** row column

Parameters:

row		Integer literal or integer variable specifying the row you want to move the cursor to. The top row of the screen is row 1. An asterisk (*) indicates that the row position does not change.
column		Integer literal or integer variable specifying the column you want to move the cursor to. The leftmost column of the screen is column 1. An asterisk (*) indicates that the column position does not change.

Reserved Variables:

$Result	Set to 0.

Example: The following example moves the cursor to column 20 of the current row.

```
Cursor Position * 20
```

The next example moves the cursor to row 15. The column position does not change.

```
Cursor Position 15 *
```

The last example uses the variables row and column to position the cursor at row 7, column 10.

```
Integer row, column

column = 10

row = 7

Cursor Position row column
```

See Also: Cursor Block, Cursor Down, Cursor Home, Cursor Left, Cursor Line, Cursor Off, Cursor Restore, Cursor Right, Cursor Save, Cursor Up

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Cursor Restore

Description: This command manipulates characters in the remote PC's display buffer. Use it to restore the position, shape (block or underscore), and video attributes of a cursor. This command is valid only for a cursor you saved using the Cursor Save command.

Syntax: **Cursor Restore**

Reserved Variables: $Result Set to 0.

Example: The following example moves the cursor to row 12, column 20; changes it to a block; then saves all the information about the cursor. Later, it restores the saved cursor.

```
Cursor Position 12 20
Cursor Block
;saves the cursor at row 12, column 20
;with block form
Cursor Save
Clear Screen
;changes the cursor's position and shape
Cursor Position 23 10
Cursor Line
;restores the cursor to row 12, column 20
;and block form
Cursor Restore
```

See Also: Cursor Block, Cursor Down, Cursor Home, Cursor Left, Cursor Line, Cursor Off, Cursor Position, Cursor Right, Cursor Save, Cursor Up

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Cursor Right

Description:	This command manipulates characters in the remote PC's display buffer. Use it to move the cursor right the specified number of columns, or right one column if no number is specified. If the number exceeds the number of columns to the right of the current position, the cursor moves to the rightmost column. The cursor remains in the same row.	
Syntax:	**Cursor Right** [count]	
Parameters:	count	(Optional) integer literal or integer variable containing the number of columns to move right. The default is 1.
Reserved Variables:	$Result	Set to 0.

Example: The following example moves the cursor right five columns.

```
Cursor Right 5
```

The next example moves the cursor right the number of columns specified in the count variable.

```
Integer count
count = 5
Cursor Right count
```

See Also: Cursor Block, Cursor Down, Cursor Home, Cursor Left, Cursor Line, Cursor Off, Cursor Position, Cursor Restore, Cursor Save, Cursor Up

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Cursor Save

Description: This command manipulates characters in the remote PC's display buffer. Use it to save the position, shape (block or underscore), and video attributes of the cursor. To restore the cursor's position and attributes, use the Cursor Restore command.

Syntax: **Cursor Save**

Reserved Variables: $Result Set to 0.

Example: The following example moves the cursor to row 12, column 20; changes it to a block; then saves all the information about the cursor. Later, it restores the saved cursor.

```
Cursor Position 12 20
Cursor Block
;saves the cursor at row 12, column 20
;with block form
Cursor Save
Clear Screen
;changes the cursor's position and shape
Cursor Position 23 10
Cursor Line
;restores the cursor to row 12, column 20
;and block form
Cursor Restore
```

See Also: Cursor Block, Cursor Down, Cursor Home, Cursor Left, Cursor Line, Cursor Off, Cursor Position, Cursor Restore, Cursor Right, Cursor Up

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Cursor Up

Description:	This command manipulates characters in the remote PC's display buffer. Use it to move the cursor up the specified number of rows, or one row if no number is specified. If the number exceeds the number of rows above the current position, the cursor moves to the top of the screen. The cursor remains in the same column.	
Syntax:	**Cursor Up** [count]	
Parameters:	count	(Optional) integer literal or integer variable containing the number of rows to move the cursor. The default is 1.
Reserved Variables:	$Result	Set to 0.

Example: The following example moves the cursor up five rows.

```
Cursor Up 5
```

The next example moves the cursor up the number of rows specified in the count variable.

```
Integer count
count = 5
Cursor Up count
```

See Also: Cursor Block, Cursor Down, Cursor Home, Cursor Left, Cursor Line, Cursor Off, Cursor Position, Cursor Restore, Cursor Right, Cursor Save

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Decrypt

Description: Use this command to decrypt a string which has been encrypted using the Encrypt command. To retrieve the original, unencoded string, you must use the exact same key string in both the Encrypt and Decrypt commands. Both the source string and key string are case-sensitive.

This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Decrypt** source key

Parameters:

source	String variable to decrypt. The string variable can contain any combination of characters and numbers.
key	String literal or string variable containing the encryption key. The key can contain any combination of characters and numbers.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

WARNING: If the wrong key is used to decrypt the data, no error is generated. The resulting string is incorrect and it overwrites the source string.

Example: The following example reads an encrypted password from a file, decrypts the password for later use, and dials CompuServe.

```
String filename[66]
String password[12]
filename = "compusrv.dat"
Find First filename
If ($Result==0) GoTo @end
Open 1 filename
Read Line 1 password
Close 1
Decrypt password "passkey"
Trim password
Dial OnlSvc "CompuServe"
;send password, etc. to CompuServe
...
@end
End
```

See Also: Encrypt

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Del

Description: This command applies to the remote PC only. Use it to delete a DOS file or group of files. It is extremely useful in deleting any temporary files you create during script processing. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Del** filename

Parameters:

filename	String literal or string variable containing the full path and filename of the file you want to delete. Wildcards (? or *) are allowed.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example deletes the file DATA.TMP.

```
Del "c:\temp\data.tmp"
```

The next example deletes all files with the extension .TMP.

```
Del "c:\temp\*.tmp"
```

The last example deletes the file or files specified by the user.

```
String filename[66]
;prompt user for a filename
Type Line "Enter the name of the file to delete: "
Input filename
;delete file
Del filename
```

See Also: Close, Copy, Create, Open, Ren, SessOpr Host Run, SessOpr Remote Run

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Delete Char

Description: This command manipulates characters in the remote PC's display buffer. Use it to delete one or more characters in the current row, beginning with and including the current cursor position. The characters in the current row to the right of the deleted characters are shifted left to fill the gap. If the last character in the row is deleted, it is replaced by a normal video space. The cursor position does not change.

NOTE: This command affects only characters in the current row. Deleting characters at the end of a row has no effect on the row beneath.

Syntax: **Delete Char** [count]

Parameters:	count	(Optional) integer literal or integer variable containing number of characters to delete. The default is 1.
Reserved Variables:	$Result	Set to 0.

Example: The following example deletes five characters to the right of the cursor.

```
Delete Char 5
```

See Also: Delete Line, Insert Char, Insert Line

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Delete Line

Description:	This command manipulates characters in the remote PC's display buffer. Use it to delete one or more rows on the screen, beginning with and including the current row, in which the cursor appears. Undeleted rows below the current row are shifted up the specified number of rows. If the last row on the screen is deleted, it is replaced by normal video spaces. The cursor is moved to column 1 of the current row.	
Syntax:	**Delete Line** [count]	
Parameters:	count	(Optional) integer literal or integer variable containing the number of rows to delete. The default is 1.
Reserved Variables:	$Result	Set to 0.
Example:	The following example deletes the five rows down from the current row. Delete Line 5	
See Also:	Delete Char, Insert Char, Insert Line	

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Description

Description: Use this command to add a brief description to your script file. You should have at most one Description command in each of your scripts.

Normally, you enter a description of your script as one of the first lines in your script file. The description is for informational purposes only.

In pcAnywhere for Windows, the description appears in the Scripts dialog box to the right of the script name after the script file is compiled.

Syntax: **Description** string

Reserved Variables: (none)

Parameters: string String literal containing a brief description of the script file.

Example: The following example describes the script's purpose.

Description "Auto logon to my office PC"

Script Editor Commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Dial Host

Description:	This command can be used for any connection type. Use it to both load the specified remote control session configuration (along with the hardware configuration it includes) and initiate a connection.	
Syntax:	**Dial Host** host_name	
Parameters:	host_name	String literal or string variable containing the name of the remote control session configuration.
Reserved Variables:	$Result	Set to 0. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, "Error Messages."
Example:	The following example calls and attempts to connect with "Office PC."	
	Dial Host "Office PC"	
See Also:	Answer, Break, Dial Number, Dial OnlSvc, Hang	

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Dial Number

Description:	Use this command to dial the specified phone numberand make a connection using the current configuration. (You can load a configuration using Load HostInfo or Load OnlSvcInfo.) Use this command when a phone number does not already appear in the configuration or if you want to override that number.	
Syntax:	**Dial Number** number	
Parameters:	number	String literal or string variable containing telephone number to dial. The maximum length of the string is 255 characters.
Reserved Variables:	$Result	Set to 0. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’
Example:	The following example loads the hardware configuration, then dials that number. Load OnlSvcInfo "ACE BBS" Dial Number "1-500-555-5555"	
See Also:	Answer, Break, Dial Host, Dial OnlSvc, Hang, Load HostInfo, Load OnlSvcInfo.	

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Dial OnlSvc

Description:	This command can be used for any connection type. Use it to both load the specified online session configuration (along with the hardware configuration it includes) and initiate a connection.	
Syntax:	**Dial OnlSvc** online_svc	
Parameters:	online_svc	String literal or string variable containing the name of the online service you are calling. The name must be as it appears in your list of online services.
Reserved Variables:	$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."	
Example:	The following example dials the online service you defined as "CompuServe."	
	Dial OnlSvc "CompuServe"	
	End Terminal	
See Also:	Answer, Break, Dial Host, Dial Number, Hang	

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Dir

Description:	This command applies to the remote PC only. Use it to display a list of a directory's files on the screen. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.	
Syntax:	**Dir** filespec	
Parameters:	filespec	String literal or string variable containing the full path or filename for the contents of the desired directory. Wildcards (? or *) are allowed. If no drive is specified, the current drive is the default.
Reserved Variables:	$Result	Set to the number of matching files found and displayed, or 0 if no files are found. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example displays all the files in the DATA directory with the extension .DAT. It also displays the number of files found.

```
String total[6]
;get directory information and set total to $Result
Dir "c:\data\*.dat"
total = $Result
Type Line "^MTotal number of files: "
Type String total        ;display total number of files
```

See Also: CD, MD, RD, SessOpr Host Run, SessOpr Remote Run

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Emulate

Description: This command applies to the remote PC only. Use it to emulate a terminal type when in terminal mode or when character sequences are output to the screen from a script. Each emulation has its own set of screen sequences to control cursor positioning, clearing the screen, screen attributes, and other behavior.

Emulate resets Load Translation to the default translation. If you load an online session configuration after the Emulate command, the emulation is reset to the one in the configuration.

Syntax: **Emulate** terminal_name

Parameters: terminal_name The name of the following predefined terminal emulations:

Terminal Emulated	**Name**
Standard ANSI Terminals	ANSI
DEC VT-100	VT100
DEC VT-102	VT102
DEC VT-220, 7-bit mode	VT220-7
DEC VT-220, 8-bit mode	VT220-8
DEC VT-52	VT52
Televideo 912	TVI912
Televideo 920	TVI920
Televideo 925	TVI925
IBM 3101	IBM3101
Data General Dasher 100	DG100
Data General Dasher 200	DG200
Wyse 50	WYSE50
Hazeltine 1500	HAZ1500
LSI ADM-3A	ADM3A
ADDS Viewpoint	ADDSVPT
X3270	X3270

Reserved Variables: $Result Set to 0. If an error occurs, set to the value of $Error.

$Error	Standard errors. See Appendix B, “Error Messages.”

Example: The following example emulates a VT-100 terminal.

```
Emulate VT100
```

See Also: Reset, Terminal

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Encrypt

Description: Use this command to convert the source string into a code based upon the key string. Both the source string and key string are case-sensitive. This command prevents unauthorized access to the information contained in the source string.

To retrieve the original, unencoded string, you must use the exact same key string in both the Encrypt and Decrypt commands. Encryption is commonly used to protect passwords and personal information. You can encrypt data and safely write it to a data file for retrieval later.

This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Encrypt** source key

Parameters:

source	String variable to encrypt. The string variable can contain any combination of characters and numbers.
key	String literal or string variable containing the encryption key. The key can contain any combination of characters and numbers.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example makes ‘‘passkey’’ the key and prompts the user for a password. After the password is encrypted, it is written to a file for later reference.

```
String password[12]
String key[12]
key = "passkey"
;prompt user for password
Type Line "Enter your password: "
;set password variable
Input password
;encrypt and decrypt password
Encrypt password key
Open 1 "compusrv.dat" "CW"
Write Line 1 password
Close 1
```

See Also: Decrypt

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

End

Description: This command ends current script processing. If the script was executed from a menu, dialog box, or window, control is returned to that menu, dialog box, or window. If the script was called by another script, control returns to the calling script. For pcAnywhere for DOS only, if the script was started from a DOS command line, pcAnywhere is terminated.

Syntax: **End** [return_code]

Parameters:

return_code	(Optional) integer literal or integer variable to return from called script to calling script. Its value is also stored in $Result.

Reserved Variables:

$Result	Set to the value of return_code so it can be used by the calling script. Set to zero when return_code is not used.
$Error	Unchanged.

Example: In the following example, the MAIN.SCR script file calls the ANOTHER.SCR script file. ANOTHER.SCR uses the global variables declared in MAIN.SCR.

```
MAIN.SCR
Integer temp
;declare $1 and $2 as global integers
Integer $1, $2
$1 = 5          ;set $1 variable
$2 = 5          ;set $2 variable
Script ANOTHER          ;call ANOTHER.SCR
temp = $Result          ;store return value in temp
Type String "The value is: "
;print product of $1 and $2 (25) to the screen
Type Line temp
Wait Time 3
End;end script processing

ANOTHER.SCR
value = $1 * $2          ;calculate product of $1 and $2
;end processing of ANOTHER.SCR,
;return control, and pass value to MAIN.SCR
End value
```

See Also: End Menu, End Terminal, Exit

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

End Menu

Description:	This command ends the current script, exits terminal mode, and returns control to the Main menu, regardless of how the script was invoked. If a connection was made, it is disconnected.
Syntax:	**End Menu**
Reserved Variables:	$Result Set to 0.
$Error	Unchanged.
Example:	The following example ends script processing and returns to the main menu. End Menu
See Also:	End, End Terminal, Exit

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

End Terminal

Description: Use this command to end all script processing and return control to the terminal mode of pcAnywhere, regardless of how the script was invoked. This allows you to continue to use pcAnywhere with your online session even though the script has ended.

Unlike the Terminal command, when terminal mode is exited, control does not return to the script containing the End Terminal command.

Syntax: **End Terminal**

Reserved Variables: (none)

Example: The following example ends script processing, but leaves you connected to your online session and leaves pcAnywhere running.

```
End Terminal
```

See Also: End, End Menu, Exit

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Exit

Description:	Use this command to end all script processing and exit pcAnywhere. Unless the Online parameter is specified, the connection is also terminated. With Online, the user can continue using the online service.	
Syntax:	**Exit** [**Online**]	
Parameters:	**Online**	(Optional) does not lower the DTR signal on exit. Essentially, if a serial connection is in progress (modem or other), this command leaves the connection alive even after you have exited the application. Used for online service serial connections only.
Reserved Variables:	(none)	
Example:	The following example ends the program and exits pcAnywhere. Exit	

The following example ends the program but keeps the connection to the online service open.

Exit Online

See Also: End, End Menu, End Terminal

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Find First

Description: This command applies to the remote PC only. Use it to search the drive for the first file that matches the file specification. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

NOTE: You can also use this command to check if a file exists; for example, before performing a Create command.

Syntax: **Find First** search_for [match_found]

Parameters:

search_for	String literal or string variable containing the name of the file you want to search for. The filename can contain wildcards (? or *). If no drive is specified, the current drive is the default.
match_found	(Optional) string variable used to store the matched filename, if found. When not used, the filename is not retrieved.

Reserved Variables:

$Result	Set to 1 if a match is found; 0 if no match is made. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example determines if a file with the .DAT extension exists in the current directory on the C: drive.

```
Find First "C:*.DAT"
;if found, go to @found
If ($Result == 1) GoTo @found
Type Line "No matching files found."
GoTo @end    ;if not found, go to @end

@found:
Type Line "Matching files found."

@end:
End;end processing
```

See Also: Dir, Find Next

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Find Next

Description: This command applies to the remote PC only. Use it to search for additional instances of the file or file types specified in the Find First command. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Find Next** [match_found]

Parameters:

match_found	(Optional) string variable used to store the matched filename, if found. When not used, the filename is not retrieved.

Reserved Variables:

$Result	Set to 1 if a match is found; 0 if no match is made. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example starts the search for files with .DAT as an extension with the Find First command. Every time a match is found, Find Next is repeated to find an additional matching file.

```
Integer count
String match[10]
;initialize count to 0
count = 0
;search for matching files
Find First "C:*.DAT" match
;if a match is found, continue at @found
If ($Result == 1) GoTo @found
;If no match is found, display message
;and terminate script
Type Line "No matching files found."
GoTo @end

@found:
Type Line "Matching files found."
count = count + 1       ;increment count by 1
Type String match       ;display filename
Type String "^M"        ;hard carriage return
;search for next instance until no more are found
Find Next match
If ($Result == 1) GoTo @found
;when no more .DAT files found, display total
Type String "Number of *.DAT files found: "
Type Line count
```

```
@end:
End;end processing
```

See Also: Dir, Find First

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Get Environment

Description: This command applies to the remote PC only. Use it to retrieve the setting for a DOS environment variable. The names of environment variables usually appear in Set statements in the AUTOEXEC.BAT file and closely resemble the directory names for the applications they represent. For example, pcAnywhere is AW, Norton Utilities is NU, and Norton AntiVirus is NAV. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Get Environment** variable value

Parameters:

variable	String literal or string variable containing the name of an environment variable.
value	String variable that will store the value of the specified environment variable. If the variable is not found, this is a null string.

Reserved Variables:

$Result	Set to 1 if a match is found; otherwise set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, “Error Messages.”

Example: The following takes the environment variable stating where Windows is installed, and puts it in the string variable “winpath”:

```
Get Environment "WinDir" winpath
```

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Get File Attr

Description: This command applies to the remote PC only. Use it to retrieve the attributes for the specified file. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Get File Attr** filename [attributes]

Parameters:

filename — String literal or string variable containing the full path and name of the file whose attributes you want. Wildcards are not allowed.

attributes — (Optional) integer variable used to store the sum of the values for the file's attributes for filename. Its value is also stored as an integer in $Result.

This values for the attributes are:

Attribute	**Value**
Read-only	1
Hidden	2
System file	4
Volume label	8
Subdirectory	16
Archive	32

For example, suppose the attributes value for the IO.SYS file is 7. Because 7 is the sum of 1, 2, and 4, so IO.SYS is read-only, hidden system file. If the sum is 0, the file is a Normal file.

Reserved Variables:

$Result — Contains the file attributes integer. A non-negative value indicates that valid attribute information has been obtained. If an error occurs, set to the value of $Error.

$Error — Standard errors. See Appendix B, "Error Messages."

Example: The following example retrieves the attributes for a file, converts each attribute to a character (such as N for normal), then displays the characters.

```
Integer attr
String description[10]
;set description to null
description = ""
;get file attributes for COMMAND.COM
Get File Attr "COMMAND.COM" attr
;if all bits are 0, write N (normal) to the string
If (attr == 0) Then StrCat description "N"
```

```
;if bit 0 is 1, write R (read-only) to the string
If ((attr & 1) == 1) Then StrCat description "R"
;if bit 1 is 1, write H (hidden) to the string
If ((attr & 2) == 2) Then StrCat description "H"
;if bit 2 is 1, write S (system file) to the string
If ((attr & 4) == 4) Then StrCat description "S"
;if bit 3 is 1,
;write V (volume label) to the string
If ((attr & 8) == 8) Then StrCat description "V"
;if bit 4 is 1, write D (directory) to the string
If ((attr & 16) == 16) Then StrCat description "D"
;if bit 5 is 1, write A (archive) to the string
If ((attr & 32) == 32) Then StrCat description "A"
Type String "The file is a "
Type Line description
```

See Also: Get File Date, Get File Size, Get File Time, Set File Attr, Set File Date, Set File Size, Set File Time

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Get File Date

Description: This command applies to the remote PC only. Use it to retrieve the date that a file was created or last modified. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Get File Date** filename [date]

Parameters:

filename	String literal or string variable containing the full path and name of the file whose date you want. Wildcards are not allowed.
date	(Optional) string variable used to store the date in the appropriate international date format (for example, YY/MM/DD for the United States). When not used, the date is not retrieved in string form. Its value is also stored as an integer in $Result.

Reserved Variables:

$Result	Stores the date as an integer. The format is YYMMDD. A non-negative value indicates that a valid date was retrieved. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, “Error Messages.”

Example: The following example finds the date for the COMMAND.COM file and stores it in $Result.

```
String filename[12]
filename = "command.com"
;get the file date for COMMAND.COM
Get File Date filename
Type Line $Result ;display date
```

See Also: Get File Attr, Get File Size, Get File Time, Set File Attr, Set File Date, Set File Size, Set File Time

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Get File Size

Description: This command applies to the remote PC only. Use it to retrieve the size, in bytes, of the file specified in filename. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Get File Size** filename [size]

Parameters:

filename	String literal or string variable containing the full path and name of the file whose size you want. Wildcards are not allowed.
size	(Optional) string variable used to store the size of the file. When not used, the size is not retrieved in string form. Its value is also stored as an integer in $Result.

Reserved Variables:

$Result	Stores the file size as an integer. Set to a non-negative value indicates a valid file size was retrieved. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example determines the size of the COMMAND.COM file.

```
String size[10]
;get the file size and display it
Get File Size "COMMAND.COM" size
Type String "The size of COMMAND.COM is: "
Type Line size
```

See Also: Get File Attr, Get File Date, Get File Time, Set File Attr, Set File Date, Set File Size, Set File Time

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Get File Time

Description: This command applies to the remote PC only. Use it to retrieve the time that the file specified in filename was created or last modified. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Get File Time** filename [time]

Parameters:

filename	String literal or string variable containing the full path and name of the file whose time you want. Wildcards are not allowed.
time	(Optional) string variable used to store the time. The time format is HHMM on a 24-hour clock. The leading zero is dropped. When not used, the time is not retrieved in string form. Its value is also stored as an integer in $Result.

Reserved Variables:

$Result	Stores the retrieved time as an integer. A non-negative value indicates that a valid time was retrieved. The time format is HHMM on a 24-hour clock (no leading zero). If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, “Error Messages.”

Example: The following example determines the time for the COMMAND.COM file and stores the time as a string in the time variable.

```
String filename[12]
String time[4]
;set filename to COMMAND.COM
filename = "command.com"
;store the time in the time variable
Get File Time filename time
```

See Also: Get File Attr, Get File Date, Get File Size, Set File Attr, Set File Date, Set File Size, Set File Time

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Get Free Disk

Description: This command applies to the remote PC only. Use it to determine the amount of space remaining on the specified disk. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Get Free Disk** drive [freespace]

Parameters:

drive	String literal or string variable specifying the drive you are polling. You can specify only one drive at a time.
freespace	(Optional) string variable containing the amount of space available (in bytes). When not used, the amount of space is not retrieved in string form. This value is also stored as an integer in $Result.

Reserved Variables:

$Result	Stores the amount of space available (number of bytes) as an integer. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example determines and displays the amount of free space on drive C.

```
String size[12]
;and get free space on drive C:
Get Free Disk C size
;display the amount of space available on drive C:
Type String "Free space on drive C: "
Type Line size
```

See Also: Dir

Script Editor Commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

GoSub

Description: Use this command to call a subroutine from within the current script file. When the subroutine ends, the script resumes processing with the line immediately following the GoSub command.

To correctly return control from the subroutine to the body of the script, each subroutine must end with a Return command. An End command should separate any subroutine or series of subroutines from the rest of the script so that the lines are executed only when a GoSub command is executed. Subroutines can be nested ten levels deep.

Syntax: **GoSub** label

Parameters:	label	A label within the current script. The label can contain numbers, letters or characters.
Reserved Variables:	$Result	Unchanged.
	$Error	Unchanged.

Example: The following example contains a subroutine called getInput that asks the user for a string.

```
String instring[255]
;call to getInput subroutine
GoSub @getInput
;if no input, end processing
If ($Result == 0) GoTo @end
Send String instring      ;send instring

@end:
End;end processing

@getInput:    ;getInput subroutine
Input instring  ;input string
StrLen instring          ;get string length of instring
Return $Result           ;return string length of instring
```

See Also: GoTo, Link, Return, Script

Script Editor Commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

GoTo

Description:	Use this command to branch, or jump, to a label within the current script file.	
Syntax:	**GoTo** label	
Parameters:	label	A label within the current script. The label can contain numbers, letters or characters.
Reserved Variables:	$Result	Set to 0.
	$Error	Unchanged.

Example: The following example uses a GoTo command to end the search for matching files when none or no more are found. For every file that is found, the script displays the filename and updates the count.

```
Integer count
String match[10]
count = 0       ;set count to 0
;search for matching files
Find First "C:*.DAT" match
;if match found, go to @found
If ($Result == 1) GoTo @found
Type Line "No matching files found."
GoTo @end    ;if not found, go to @end

@found:
Type Line "Matching files found."

@foundmore:
count = count + 1          ;increment count by 1
Type String match          ;display filename
Type String "^M"           ;hard carriage return
;search for next instance until no more are found
Find Next match
If $Result == 1 GoTo @foundmore
Type String "Number of *.DAT files found: "
Type Line count            ;display count

@end:
End;end processing
```

See Also: GoSub, Link, Return, Script

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Hang

Description: Use this command to terminate any connection.

Syntax: **Hang**

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.	
$Error	Standard errors. See Appendix B, "Error Messages."	

Example: The following example connects to a host PC, then later terminates that connection.

```
Dial Host "Office PC"
...
Hang            ;terminate connection
```

See Also: Answer, Break, Dial Host, Dial Number, Dial OnlSvc

Script Editor Commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

If...Then

-or-

If...GoTo

Description: Use this command for conditional processing. If the specified expression is true, the command is executed. If the expression is false, the command is ignored and script processing continues with the next command.

Syntax: **If** expression **Then** command

Or,

If expression **GoTo** label

Parameters:

expression	Logical expression that results in a true or value. Expressions cannot be used to compare values in strings containing both characters and numbers, because the string variables are converted to their numeric equivalents. For example, the string "123ABX" has the numeric value of 123.
label	A label within the current script. The label can contain numbers, letters or characters.
command	Any valid script command.

Reserved Variables:

$Result	Unchanged.
$Error	Unchanged.

Example: The following example uses If...GoTo commands to control the search for matching files. When a file is found, the search continues by going to the found or foundmore label. For every matching file, the script displays the filename and updates the count.

```
Integer count
String match[10]
count = 0       ;set count to 0
;search for matching files
Find First "C:*.DAT" match
;if match found, go to @found
If ($Result == 1) GoTo @found
Type Line "No matching files found."
GoTo @end    ;if not found, go to @end

@found:
Type Line "Matching files found."

@foundmore:
count = count + 1       ;increment count by 1
Type String match       ;display filename
Type String "^M"        ;hard carriage return
```

```
;search for next instance until no more are found
Find Next match
If $Result == 1 GoTo @foundmore
Type String "Number of *.DAT files found: "
Type Line count          ;display count

@end:
End;end processing
```

See Also: GoTo

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Index

Description: Use this command to locate the starting position of one string within another. The first character of a string is at position 1. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Index** source_str search_str [position]

Parameters:

source_str	String literal or string variable to be searched.
search_str	String literal or string variable you want to search for.
position	(Optional) integer variable used to store the starting position of the first instance of the search string. Its value is also stored in $Result.

Reserved Variables:

$Result	Stores the starting position of the first instance of the search string. If no match is found, set to 0.
$Error	Unchanged.

Example: The following example finds the starting position of a substring within a string.

```
String source[20]
Integer position
position = 2
source = "ABCDEFGHIJ"
;search for FGH in source
;and store the result in position
Index source "FGH" position
```

See Also: Lower, Set IgnoreCase, StrCat, StrCmp, StrLen, StrSet, SubStr, Trim, Upper

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Input

Description: Use this command to retrieve data entered by the user from the keyboard. Input retrieves all the characters entered before the Enter key, up to the specified maximum length, or up to the length of the destination string. The user can use any of the following keys to edit the entry before pressing Enter:

Keystroke	**Operation**
LeftArrow	Moves the cursor one column to the left
RightArrow	Moves the cursor one column to the right
Backspace	Deletes the character before the cursor
Return	Terminates data input

This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Input** data_string [maxlength]

Parameters:

data_string		String variable where the retrieved data is stored.
maxlength	(Optional) integer literal or integer variable containing the maximum number of characters accepted into the string. The default value is the declared length of the data_string.	

Reserved Variables:

$Result		Contains the total number of characters entered (excluding the terminating carriage return). If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, “Error Messages.”	

Example: The following example retrieves the user’s response to "Enter your name...".

```
String username[30]
Type Line "Enter your name (up to 30 characters): "
;retrieve the name typed by the user
;and store it in the username variable
Input username 30
Send Line username      ;send username
```

See Also: Input Key, Keyboard Flush, Keyboard Hit

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Input Key

Description: Use this command to retrieve a single character typed by the user from the keyboard. The entry is not echoed to the screen and the user does not have to press Enter. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Input Key** character

Parameters:

character	String variable where the retrieved character is stored.

Reserved Variables:

$Result	Stores the ASCII value of the keystroke entered.
$Error	Unchanged.

Example: The following example asks the user for a Y or an N for yes or no until one is received. If the response (stored in the yn variable) is a Y, the script continues. If it is an N, the script ends.

```
String yn[1]
Type Line "Enter Y to continue; N to exit: "

;loop until the user presses y or n
@loop:
Input Key yn
Upper yn        ;convert user input to uppercase
;compare user input with "Y" and "N"
StrCmp yn "Y"
If ($Result == 0) GoTo @yes
StrCmp yn "N"
If ($Result == 0) GoTo @no
;beep and keep looping if user presses
;key other than y or n
Beep
GoTo @loop

@no:
Exit

@yes:
;continue processing the remainder of the script
...
```

See Also: Input, Keyboard Flush, Keyboard Hit

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Insert Char

Description:	Use this command to declare the number of characters to be inserted at the current cursor position, then use Type String to insert the desired characters. Insert Char is always used in conjunction with the Type String command. The character originally marked by the cursor and the rest of the characters in that row are shifted to the right. Characters that are shifted off the screen are lost. The cursor position does not change.	
Syntax:	**Insert Char** [count]	
Parameters:	count	(Optional) integer literal or integer variable containing number of characters to be inserted. The default is 1.
Reserved Variables:	$Result	Set to 0.
$Error	Unchanged.	
Example:	The following example moves the cursor to an appropriate position and inserts a message in a previously typed line. Type string “This is a test.” Cursor Left 10 Wait Time 2 ;prepare for insertion Insert Char 21 ;insert the characters Type String "Waiting for signal..."	
See Also:	Delete Char, Delete Line, Insert Line	

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Insert Line

Description:	Use this command to insert one or more rows of normal video spaces at the row containing the cursor. The row originally containing the cursor and all rows beneath it are shifted down. Rows that are shifted off of the screen are lost. The row position for the cursor does not change, but it is moved to column 1 (if it was not already there).	
Syntax:	**Insert Line** [count]	
Parameters:	count	(Optional) integer literal or integer variable containing number of rows to insert. The default is 1.
Reserved Variables:	$Result	Set to 0.
	$Error	Unchanged.
Example:	The following example inserts three rows above the current row.	
	Integer count	
	count = 3	
	Insert Line count	;insert 3 rows at current row
See Also:	Delete Char, Delete Line, Insert Char	

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Keyboard Flush

Description: Use this command to clear the keyboard buffer containing any characters that the user may have typed ahead. Immediately before retrieving keyboard input (using Input or Input Key), use Keyboard Flush to clear the keyboard buffers. This ensures that you retrieve only the characters the user types in response to a prompt. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Keyboard Flush**

Reserved Variables: $Result Set to 0.

$Error Unchanged.

Example: The following example clears the keyboard buffer prior to accepting a filename typed by the user.

```
String filename[12]
Type Line "Enter the name of the file to delete: "
Keyboard Flush          ;clear the keyboard buffer
Input filename;set the filename variable
Del filename   ;delete the file
```

See Also: Input, Input Key, Keyboard Hit

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Keyboard Hit

Description: Use this command to determine if a character has been entered, but not yet read using an Input or Input Key command. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Keyboard Hit** [flag]

Parameters:

flag	(Optional) integer variable used as a flag indicating whether or not a key has been pressed. Becomes a 1 if a key has been pressed; becomes a 0 otherwise. Its value is also stored in $Result.

Reserved Variables:

$Result	Set to 1 if a key has been pressed; 0 otherwise.
$Error	Unchanged.

Example: The following example determines if the user has pressed Esc.

```
@checkesc:   ;beginning of subroutine
Keyboard Hit  ;check for keystroke
;if not, go back to calling routine
If ($Result == 0) then Return 0
Input Key inkey          ;if yes, input keystroke
StrCmp inkey "^["        ;is it the Esc key?
;if yes, Return a value of 1
If ($Result == 0) then Return 1
;repeat loop until all keys are entered and checked
GoTo @checkesc
```

See Also: Input, Input Key, Keyboard Flush

Script Editor Commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Let

Description:	Use this command to assign an expression to a variable. You do not need to use the reserved word Let unless the variable name you are using is identical to a reserved word, such as a command name.
Syntax:	[**Let**] variable = expression
Parameters:	**Let** (Optional) reserved word; the command may be shortened.
variable	Integer variable or string variable.
expression	Any expression containing integers, strings, and operators.
Reserved Variables:	(none)
Example:	The following example assigns a string literal to the echo variable and numeric expressions to the variables named base and number.

```
String echo[10]
Integer base, number
Let echo = "ABC"        ;set echo to the string ABC
base = 1
number = base * 10
```

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Link

Description: Use this command to terminate one script, losing all its local variables and their values, and start another. Control is transferred to the script named in the command. A script called using the Link command can be thought of as a continuation of the original calling script, except that the original script's local variables are lost.

This differs from the Script command in that control returns to the calling script after the called script terminates.

For example, consider an original calling script named MAIN.SCR. This script is responsible for all processing during a session. At one point, an item needs to be printed. Using the Script command, MAIN calls a secondary script, CHSPRNT.SCR, which determines which print routine to call (see Figure 4-2). CHSPRNT uses a Link command to call the routine for the selected printer. When the item has been sent to the printer, the linked script terminates and control is returned to MAIN.SCR (bypassing CHSPRNT.SCR entirely).

Figure 4-2 Once the selected print routine has finished executing, control returns to MAIN.SCR.

Syntax: **Link** script_name

Parameters:	script_name	String literal or string variable containing the name of the script file to link to. The string must not contain the .SCR file extension.
Reserved Variables:	$Result	Set to 0.
	$Error	Unchanged.

Example: The following example ends one script and starts another to which the first script has been linked.

```
Type Line "This script passes control to another."
Link "Script2"
```

See Also: GoSub, GoTo, Return, Script

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Load FKeys

Description: Use this command to load a function key file containing programmable function key information. Function key translations are in effect only in terminal mode (not during script execution).

Syntax: **Load FKeys** keyboard_file

Parameters:

keyboard_file	String literal or string variable containing name of function key file. Do not include a filename extension (.MK6 is assumed).

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example loads the remote control session configuration and function keys information for the host PC named MCI Mail, then dials the host.

```
;load remote control session configuration
Load HostInfo "MCI Mail"
;load function key table for use with MCI
Load FKeys "MCI-MAIL"
;call MCI Mail
Dial Number "9,555-5555"
End Terminal  ;go to terminal mode
```

See Also: Dial Host, Dial Number, Dial OnlSvc, Load HostInfo, Load OnlSvcInfo, Load Translation

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Load HostInfo

Description:	Use this command to load a remote control session configuration. The hardware configuration associated with the specified remote control session configuration is automatically loaded at the same time. This command is used primarily when you need to prompt the user for or override some information in the remote control session configuration. If you don't want to change any settings, you may prefer to use Dial Host.	
Syntax:	**Load HostInfo** host_name	
Parameters:	host_name	String literal or string variable containing the name of the remote control session configuration to load.
Reserved Variables:	$Result	Set to 0. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, "Error Messages."
Example:	The following example loads the remote control session configuration named Office PC, then dials the host.	

```
;load remote control session configuration that has
;no phone number
Load HostInfo "Office PC"
;attempt to connect with office PC
Dial Number "555-5555"
```

See Also: Dial Host, Dial Number, Dial OnlSvc, Load FKeys, Load OnlSvcInfo, Load Translation

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Load OnlSvcInfo

Description: Use this command to load the specified online session configuration. The hardware configuration associated with the specified online session configuration is automatically loaded at the same time. This command is used primarily when you need to prompt the user for or override some information in the online session configuration. If you don't want to change any settings, you may prefer to use Dial OnlSvc.

Syntax: **Load OnlSvcInfo** online_svc_name

Parameters:

online_svc_name	String literal or string variable containing the name of the online session configuration to load. The name must be as it appears in the pcAnywhere list of online services.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example loads the online session configuration, asks the user for the phone number, then dials that number.

```
String phone_number[12]
;load CompuServe session configuration
;that has no phone number
Load OnlSvcInfo "CSwoPHONE"
Type Line "Enter CompuServe phone number: "
Input phone_number 12
;attempt to connect with online service
Dial Number phone_number
```

See Also: Dial Host, Dial Number, Dial OnlSvc, Load FKeys, Load HostInfo, Load Translation

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Load Translation

Description: Use this command to load a translation table file for both incoming and outgoing character streams. A translation table is used to translate one character set to another or to filter out unwanted characters. You must load the configuration before the Load Translation command is executed. The Set Translation command indicates whether or not to use the loaded translation file.

Syntax: **Load Translation** translation_file

Parameters:

translation_file	String literal or string variable containing the translation filename to load. Do not include the .TRN filename extension.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, “Error Messages.”

Example: The following example loads the strip high bits translation table, loads a remote control session configuration, enables the translation, and calls the host. The order of these commands must be as shown.

```
Load OnlSvcInfo "ACE BBS"        ;load configuration
Load Translation "STRIPHIG"
Set Translation On
...
Dial Number ACS_BBS  ;attempt connection
```

See Also: Load FKeys, Load HostInfo, Load OnlSvcInfo, Set Translation

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Lower

Description:	Use this command to convert all uppercase characters in a string to lowercase. Characters that are already lowercase are not affected. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.	
Syntax:	**Lower** string	
Parameters:	string	String variable to convert.
Reserved Variables:	$Result	Set to 0. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example converts the specified string to all lowercase characters.

```
String mixed[10]
mixed = "aBC"
Lower mixed
;display string (abc is displayed)
Type String mixed
```

See Also: Set IgnoreCase, StrCat, StrCmp, StrLen, StrSet, SubStr, Trim, Upper

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

MD

Description: This command applies to the remote PC only. Use it to create a new directory. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **MD** dirname

Parameters:

dirname	String literal or string variable containing the full path and name of the directory you want to create.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example creates a new subdirectory named AW.

```
MD "AW"
```

The next example asks the user for the name of the new subdirectory.

```
String newdir[66]
Type Line "Enter the directory: "
;read path into the newdir variable
Input newdir 66
;make the directory based upon the newdir variable
MD newdir
```

The last example creates a new subdirectory using a pathname read from a file.

```
String newdir[66]
Open 1 "datafile.txt"       ;open the data file
Read String 1 newdir 66;read in the path string
Close 1         ;close the data file
;make the directory based upon the newdir variable
MD newdir
```

See Also: CD, Dir, RD, SessOpr Host Run, SessOpr Remote Run

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Message Box

Description: Use this command to display a message box. The value you use for the options parameter dictates the appearance of the message box. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **MessageBox** title text [options]

Parameters:

title — String literal or string variable containing the title of the message box.

text — String literal or string variable containing the text to appear in the message box.

options — (Optional) integer literal or integer variable which is the sum of the numbers representing the buttons to appear in the message box, the button that is selected by default, and (in Windows only) the icon to display to the left of the text.

When options is not specified in DOS, the default is 0—an OK button which is selected by default; in Windows, the default is 21—the Retry and Cancel buttons (with Retry selected by default) and a Stop icon.

Values for the options parameter are:

Value	Buttons to Display
0	OK
1	OK and Cancel
2	Abort, Retry, and Ignore
3	Yes, No, and Cancel
4	Yes and No
5	Retry and Cancel

Value	Default Button
0	Button 1
256	Button 2
512	Button 3

Value	Icon to Display
0	None
16	Stop sign

32	Question mark
48	Exclamation point
64	Information

Reserved Variables: $Result Set to one of the following values depending on which button the user pressed:

$Result	**Button Pressed**
1	OK
2	Cancel
3	Abort
4	Retry
5	Ignore
6	Yes
7	No

$Error Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example produces a message box. In pcAnywhere for Windows, the message box is as shown in Figure 4-3.

```
Session Dial "Office PC"
String message
String filename
message = "Do you want to view the file now?"
filename = "c:\excel\excel.exe 4th_qtr.xls"
Type Line "Launching application on host..."
MessageBox "View File" message 35
If ($Result == 6) Then SessOpr Host Run filename
```

Figure 4-3 The 35 in the MessageBox command gives this Windows message box its question mark as well asthe Yes, No, and Cancel buttons.

See Also: Type Line, Type String

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

On Cancel

Description: Use On Cancel to specify a command to be executed if the user presses Esc during the execution of a Wait or Receive command. If the previous Set Cancel command was set to on and the user presses Esc, the command following On Cancel is executed. If Set Cancel was not executed or set to off, On Cancel is ignored.

Syntax: **On Cancel** [command]

Parameters:

command	(Optional) command to execute if Esc is pressed. If no command is specified, then $Cancel is set to 0 and On Cancel processing is disabled (same effect as executing Set Cancel Off).

Reserved Variables:

$Result	Unchanged.
$Error	Unchanged.

NOTE: The command that is executed may set $Result and $Error.

Example: The following example enables the Cancel operation, then specifies what to do when the user cancels (by pressing Esc).

```
Set Cancel On          ;enable Cancel operation
;specify where to branch when Esc pressed
On Cancel GoTo @cancel
;dial an online service
Dial OnlSvc "CompuServe"
;wait for a specific string
Wait String "Enter password"
...
;control goes here when Esc pressed
@cancel:
Type String "^M^JOperator canceled"
```

See Also: On Disconnect, On Error, On Receive, On Timeout, Set Cancel

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

On Disconnect

Description: Use On Disconnect to specify the command to execute if a connection is lost. If the previous Set Disconnect command was set to on and the connection is lost, the command following the On Disconnect command is executed. If Set Disconnect was not executed or set to off, On Disconnect is ignored.

Syntax: **On Disconnect** [command]

Parameters:

command	(Optional) command to execute if the connection is lost. If no command is specified, then $Disconnect reserved variable is set to 0 and On Disconnect processing is disabled (same effect as executing Set Disconnect Off).

Reserved Variables:

$Result	Unchanged.
$Error	Unchanged.

NOTE: The command that is executed may set $Result and $Error.

Example: The following example enables the Disconnect operation, then specifies what to do when the connection is lost.

```
Set Disconnect On       ;enable disconnect operation
;specify where to branch when connection is lost
On Disconnect GoTo @disconnect
;attempt to connect with CompuServe
Dial OnlSvc "CompuServe"
Wait Time 2
;send Ctrl+C to alert online service
Send Char 3
;end script processing and return to terminal mode
End Terminal

;control branches here when connection is lost
@disconnect:
End
```

See Also: On Cancel, On Error, On Receive, On Timeout, Set Disconnect

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

On Error

Description: Use On Error to specify the command to execute when the $Error reserved variable is set to a non-zero value by another command. It applies to errors that occur within the same script and as a result of commands executed after the On Error command is executed. Each nested script must declare its own On Error as appropriate.

The On Error command does not reset $Error to zero before the next command is executed. You should do that to avoid reprocessing one error over and over.

Syntax: **On Error** [command]

Parameters:

command		(Optional) command to execute if an error occurs. If no command is specified, then On Error execution is disabled.

Reserved Variables:

$Result		Unchanged.
$Error		Unchanged.

NOTE: The command that is executed may set $Result and $Error.

Example: In the following example, if an error occurs between the On Error command and the End command, script control transfers to the label named @error.

```
$Error = 0      ;set $Error to 0
;specify where to branch when an error occurs
On Error GoTo @error
;attempt to connect with CompuServe
Dial OnlSvc "CompuServe"
;end script processing and return to terminal mode
End Terminal
Wait Carrier
;send carriage return to alert CompuServe
Send Char 13

@end:
End;end script processing

@error:         ;prints messages to screen
Type Line "An error has occurred.^M^J"
Type Line "Ending script processing."
GoTo @end
```

See Also: On Cancel, On Disconnect, On Receive, On Timeout

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

On Receive

Description: Use On Receive to specify a command to execute if the specified string is received during the execution of a Wait or Receive command. If the string is not received, or an incorrect string is received, the On Receive command is disabled. This command is most useful for automatic transmission of passwords and other codes.

Syntax: **On Receive** string [command]

Parameters:

string	String literal or string variable containing the characters that, if received, cause the command to execute.
command	(Optional) command to execute if string is received. To disable On Receive processing, don't specify a command.

Reserved Variables:

$Result	Unchanged.
$Error	Unchanged.
$RXString	Stores specified string.

NOTE: The command that is executed may set $Result and $Error.

Example: The following example enables On Receive and specifies a GoSub command to execute if "Enter password: " is received.

```
On Receive "Enter password: " GoSub @sendPW
;wait for this string, but if "Enter password:" is received, then branch to subroutine
Wait String "Enter selection: "
End;end script processing

@sendPW:
;send password followed by carriage return
Send String "acx14^M"
Return          ;return control to calling point
```

See Also: On Cancel, On Disconnect, On Error, On Timeout

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

On Timeout

Description: Use On Timeout to specify a command to be executed if nothing is entered or received in the time allotted and a timeout occurs. If the time limit set previously by a Set Timeout command expires, the command following the On Timeout command is processed. If Set Disconnect was not executed or set to Off, the On Timeout command is ignored.

Syntax: **On Timeout** [command]

Parameters:

command	(Optional) command to execute if the allotted time expires. If command is not specified, then On Timeout processing is disabled and the $Timeout reserved variable is set to 0.

Reserved Variables:

$Result	Unchanged.
$Error	Unchanged.

NOTE: The command that is executed may set $Result and $Error.

Example: The following example sets a Timeout value of 60 seconds. The On Timeout command indicates what the script will do if that time limit is exceeded.

```
Set Timeout 60          ;set the default timeout value
;specify where to branch if timeout occurs
On Timeout GoTo @timeout
;wait for a specific string
Wait String "Enter password: "
...
;control goes here if timeout occurs
@timeout:
Type Line "Timeout!"
Type Line "Exiting..."
Exit
```

See Also: On Cancel, On Disconnect, On Error, On Receive

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Open

Description: This command applies to the remote PC only. Use it to open an existing file for reading or writing. No more than nine files can be open at the same time. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

To prevent "File not found" errors from occurring, specify a "C" attribute in addition to other attributes. This flag tells pcAnywhere to create the file if it doesn't exist and prevents script processing from interruption. This is especially helpful during unattended sessions.

Syntax: **Open** file_number filename [attributes]

Parameters:

file_number	Integer literal used to identify the file. Valid identifiers are 1 through 9.
filename	String literal or string variable containing the name of the file to open.
attributes	(Optional) string literal or string variable used to set the attributes of the file. The default is non-sharable read/write.

The following are valid attribute settings:

R	Read-only
W	Read/write
S	Sharable
C	Create file if it doesn't already exist

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example opens FILE1.TXT as a sharable read/write file in the current directory. The file must already exist.

```
Open 1 "file1.txt" "SW"
Or,
String filename[12]
;assign the filename to the string variable
filename = "file1.txt"
;create the new file with read/write attributes
Open 1 filename "CW"
```

See Also: Close, Copy, Create, Del, Ren

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Print File

Description:	Use this command to print a file to the remote PC's current printer. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.	
Syntax:	**Print File** filename	
Parameters:	filename	String literal or string variable containing the full path and filename of the file to print.
Reserved Variables:	$Result	Set to 0. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, "Error Messages."
Example:	The following example prints TEST.SCR using the current printer.	
	String filename[80]	
	filename = "c:\aw\test.scr"	
	Print File filename	;print the file
See Also:	Print Line, Print String, Printer	

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Print Line

Description:	Use this command to print a string to the remote PC's current printer. Once the string has been printed, the contents of the reserved variable $TLineEnd are also printed. The default value for $TLineEnd is a single carriage return and a single linefeed (^M^J). This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.	
Syntax:	**Print Line** string	
Parameters:	string	String literal or string variable containing the information to print.
Reserved Variables:	$Result	Set to number of characters output. The quotation marks surrounding the string are not included in the count, but the contents of $TLineEnd are. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."	
Example:	The following example prints "abc" followed by the contents of $TLineEnd. Print Line "abc"	
See Also:	Print File, Print String, Printer	

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Print String

Description:	Use this command to print a string to the remote PC's current printer. The Print String command differs from Print Line in that $TLineEnd is not appended to the end of the string. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.	
Syntax:	**Print String** string	
Parameters:	string	String literal or string variable containing the information to print.
Reserved Variables:	$Result	Set to number of characters output. The quotation marks surrounding the string are not included in the count. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, "Error Messages."
Example:	The following example prints "abc" followed by a carriage return and a linefeed. Print String "abc^M^J"	
See Also:	Print File, Print Line, Printer	

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Printer

Description: Use this command to turn simultaneous printing on and off. When on, data that is sent to the screen is also sent to the remote PC's current printer. This includes screen sequences used to perform special functions, such as repositioning the cursor or setting screen attributes. Simultaneous printing occurs in the following situations:

- Data received when after a Set Echo On command
- Output from a File, Type Line, and Type String commands
- Data received in terminal mode after a Printer On command

This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Printer On** | **Off**

Parameters:

On	Toggles simultaneous printing on.
Off	Toggles simultaneous printing off.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example turns simultaneous printing on, sends data to the printer, then turns the printing off.

```
Printer On
Type Line "This is sent to the printer."
Type Line "^M^J"
Printer Off
Type Line "This is not sent to the printer."
```

See Also: Print File, Print Line, Print String, Set Echo, Type File, Type Line, Type String

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

RD

Description: This command applies to the remote PC only. Use it to remove an empty directory. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **RD** dirname

Parameters:

dirname	String literal or string variable containing the full name and path of the directory you want to remove.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	If the directory is not empty, $Error is set to -33. See Appendix B, "Error Messages."

Example: The following example removes the TEMP directory.

```
RD "C:\TEMP"
```

The next example prompts the user for a directory to remove.

```
String deldir[66]
Type Line "Enter the directory: "
;read path (input by user) into the deldir variable
Input deldir 66
;remove the directory
RD deldir
```

The last example removes a directory whose path is read from a file.

```
string deldir[66]
;open the data file, read a path from it,
;and close the file
Open 1 "DATAFILE.TXT"
Read String 1 deldir 66
Close 1
;remove the directory
RD deldir
```

See Also: CD, Dir, MD, SessOpr Host Run, SessOpr Remote Run

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Read Line

Description: This command applies to the remote PC only. Use it to read a line from a file and store it in a variable. By default, a line is terminated by an end-of-line character. You can specify the line length to a maximum of 255 characters. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

NOTE: A terminating character of 0xFF (255 in decimal) cannot be used.

Syntax: **Read Line** file_number data_string [length [termchar]]

Parameters:

- file_number — Integer literal used by Open or Create command to identify the file. Valid identifiers are 1 through 9.
- data_string — String variable where the data is stored. The terminating characters are not stored.
- length — (Optional) integer literal or integer variable specifying the maximum number of characters to read. The default value is the declared length of data_string. Termination characters are not included.
- termchar — (Optional) A literal or variable (that can be either an integer or string type) specifying the line's terminator. The default terminator is the contents of $RLineEnd (whose default value is a carriage return/linefeed). A terminating character is expressed as its ASCII decimal equivalent. (For example, a carriage return is expressed as ASCII decimal 13.) To specify a terminating character, you must also specify the length parameter. To specify more than one terminating character, use a string literal or string variable and separate the ASCII values for the characters with a space.

Reserved Variables:

- $Result — Set to the number of characters read. If an error occurs, set to the value of $Error. When an end-of-file marker is encountered, $Result is set to -12.
- $Error — Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example opens a data file, reads a line containing a path from it, and closes the file. Then the script changes the current directory to that path.

```
string newpath[66]
Open 1 "datafile.txt"
Read Line 1 newpath 66
Close 1
;change directory to the newpath variable
CD newpath
```

See Also: Read String, Write Line, Write String

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Read String

Description: This command applies to the remote PC only. Use it to read a string of characters from a file and store it in a variable. Each read command advances the file pointer. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Read String** file_number data_string [length]

Parameters:

file_number	Integer literal used by Open or Create command to identify the file. Valid identifiers are 1 through 9.
data_string	String variable which stores the data. The terminating characters are not stored.
length	(Optional) integer literal or integer variable specifying the maximum number of characters to read. The default value is the declared length of data_string.

Reserved Variables:

$Result	Set to the number of characters read. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, “Error Messages.”

Example: The following example opens the data file, reads its first 80 characters, its next 52 characters, then closes the file.

```
String data1[80]
String data2[80]
Open 1 "datafile.txt"
Read String 1 data1 80
Read String 1 data2 52
Close 1
```

See Also: Read Line, Write Line, Write String

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Receive Char

Description: Use this command to receive a single character from the communications port and store it in a variable.

Syntax: **Receive Char** data [timeout]

Parameters:

data	Integer variable or string variable where the received character is stored. If data is a string variable, the character received is stored in the first position in the string.
timeout	(Optional) integer defining a timeout value (in seconds). If this parameter is omitted, the current $Timeout value is used.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example retrieves and displays characters until six seconds goes by without a character being typed.

```
String char[1]
Set Timeout 6 ;6 second time limit set
;where to branch when time expires
On Timeout GoTo @end

@loop:
Receive Char char      ;receive a character into char
Type String char       ;display char
GoTo @loop

@end:
;on timeout display message
Type Line "Timeout!"
Exit
```

See Also: Set Cancel, Set Disconnect, Set Timeout, Send commands, other Receive commands

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Receive Clear

Description:	Use this command to clear the receive buffer of any data not yet read using Receive commands.
Syntax:	**Receive Clear**
Reserved Variables:	$Result Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’
Example:	The following example clears the receive buffer and receives a string into the data variable.
	String data[60]
	Receive Clear
	Receive String data 30
See Also:	Set Cancel, Set Disconnect, Set Timeout, Send Commands, other Receive commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Receive File

Description:	Use this command to receive a file using the protocol previously specified by a Set Protocol command. During the transfer, a status screen displays the transfer's progress.	
Syntax:	**Receive File** [filename]	
Parameters:	filename	(Optional) string literal or string variable containing the path and filename where the incoming file will be stored. Some protocols do not allow you to specify a filename, since part of the protocol dictates that the sender informs the receiving machine what filename to use.
Reserved Variables:	$Result	Set to a non-negative integer value. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, "Error Messages."
Example:	The following example performs a file transfer using the XMODEM protocol. Set Protocol XMODEM Receive File "c:\data\datafile.txt"	
See Also:	Set Cancel, Set Disconnect, Set Timeout, Send Commands, other Receive commands	

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Receive Line

Description: Use this command to receive a string from the communications port and store it in a variable. The Receive Line command is terminated when one of the following occurs:

- The string has been filled
- A line terminating sequence has been specified: either termchar (if it has been specified) or the characters equal to $RLineEnd (if termchar has not been specified)
- The elapsed time is equal to the specified timeout value

NOTE: A terminating character of 0xFF (255 decimal) is not allowed.

Syntax: **Receive Line** data_string [length [termchar [timeout]]]

Parameters:

data_string	String variable used to store the received data. The terminating characters are not stored.
length	(Optional) integer literal or integer variable specifying the maximum number of characters to read. The default value is the declared length of data_string. Termination characters are not included.
termchar	(Optional) A literal or variable (that can be either an integer or string type) specifying a line terminator. The default terminator is the contents of $RLineEnd (whose default value is a carriage return/linefeed). A terminating character is expressed as its ASCII decimal equivalent. (For example, a carriage return is expressed as ASCII decimal 13.) To specify a terminating character, you must also specify the length parameter. To specify more than one terminating character, use a string literal or string variable and separate the ASCII values for the characters with a space.
timeout	(Optional) integer defining a timeout value (in seconds). If this parameter is omitted, the current $Timeout value is used. To specify a timeout value, you must also specify values for both length and termchar.

Reserved Variables:

$Result	Set to the length of the received string. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example receives the first 80 characters of the line into the data variable.

```
String data[80]
Receive Line data
```

The next example receives characters into the data variable until the first of three situations occur: 50 characters have been received, a carriage return has been encountered, or 20 seconds have elapsed.

```
String data[80]
Receive Line data 50 13 20
```

See Also: Set Cancel, Set Disconnect, Set Timeout, Send Commands, other Receive commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Receive String

Description: Use this command to receive a string from the communications port and store it in a variable. Input is terminated when the string is full, when the specified length has been read or when the timeout value has elapsed.

Syntax: **Receive String** data_string [length [timeout]]

Parameters:

data_string	String variable used to store the received data. The terminating characters are not stored.
length	Integer literal or integer variable specifying the maximum number of characters to read. The default value is the declared length of data_string.
timeout	(Optional) integer defining a timeout value (in seconds). If this parameter is omitted, the current $Timeout value is used. If you specify a timeout value, you must also specify a value for length.

Reserved Variables:

$Result	Set to the length of the received string. If an error occurs, set to the value of $Error.
$Error	The error value for a timeout is -4. See Appendix B, "Error Messages."

Example: The following example receives the first 30 characters into the data variable.

```
String data[80]
Receive String data 30
```

The following example receives 20 characters into the data variable or times out after 10 seconds.

```
String data[80]
Receive String data 20 10
```

See Also: Set Cancel, Set Disconnect, Set Timeout, Send Commands, other Receive commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Ren

Description: This command applies to the remote PC only. Use it to change the name of a DOS file or directory. The Ren command moves a file from one directory to another if you specify a different pathname, but cannot be used to move a file from one drive to another. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Ren** old_name new_name

Parameters:

old_name	String literal or string variable containing the original path and filename or directory.
new_name	String literal or string variable containing the new name of the file or directory. The new name must not be the name of an existing file or directory.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example renames DRAFT1.TXT to DRAFT2.TXT.

```
Ren "draft1.txt" "draft2.txt"
```

The next example prompts the user for the new name of a file.

```
String newname[12]
Type Line "Enter the name of the new file: "
Input newname
Ren "draft1.txt" newname
```

See Also: Close, Copy, Create, Del, Open, SessOpr Host Run, SessOpr Remote Run

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Reset

Description: Use this command to reset the terminal emulation screen displayed on the remote PC. The screen is cleared of all characters, and the video attribute is cleared and reset to normal. All special modes, such as simultaneous print and monitor, are set to off. The cursor is moved to row 1, column 1.

Syntax: **Reset**

Reserved Variables: $Result Set to 0.

$Error Unchanged.

Example: The following example resets the terminal emulation screen. You may want to use it between calls to different online services.

```
Reset ;reset terminal
```

See Also: Emulate, Terminal

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Return

Description: Use this command to terminate the processing of a subroutine. Control is passed back to the line immediately following the GoSub command that initiated the subroutine.

Syntax: **Return** [return_code]

Parameters:

return_code	(Optional) integer literal or integer variable passed back to the calling script. Its value is also stored in $Result.

Reserved Variables:

$Result	Set to return_code. If return_code is not used, set to 0.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The subroutine in the following example asks the user for a string, then determines and returns the length of that string. After the subroutine terminates, the string is sent through the communications port (unless its length is 0).

```
String instring
Integer len
GoSub @getInput         ;call getInput subroutine
;if no input, end processing
If (len == 0) GoTo @end
Send String instring    ;send instring

@end:
End

@getInput:    ;begin getInput subroutine
Input instring  ;retrieve user entry string
;get string length of instring
StrLen instring len
;return string length of instring to caller
Return len
```

See Also: GoTo, GoSub, Link, Script

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Run

Description: This command applies to the remote PC only. Use it to execute a DOS or Windows command line on the remote PC.

NOTE: An application may not be able to run if there is not sufficient memory available.

Syntax: **Run** command_line

Parameters:

command_line	String literal or string variable containing the full path and application to execute. If no path is specified, the current directory and the PATH is searched. In addition to the executable filename, you can specify parameters.

Reserved Variables:

$Result	Set to the DOS errorlevel.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example runs EDIT.COM.

```
Run "c:\dos\edit.com"    ;run DOS editor
```

In the Windows environment, the next example loads AUTOEXEC.BAT into NOTEPAD.EXE, ready for editing.

```
String launch[40]
launch = "c:\windows\notepad.exe c:\autoexec.bat"
;launch Windows Notepad and load autoexec.bat file
Run launch
```

See Also: End, End Menu, End Terminal, Exit

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Screen Restore

Description: This command applies to the remote PC's display buffer. Use it to restore a previously saved screen image, including cursor position and attributes. The Screen Restore command is ignored if no image has been saved.

Syntax: **Screen Restore**

Reserved Variables: (none)

Example: The following example saves the contents of the screen and later restores those contents.

```
;move the cursor to row 12, column 20
Cursor Position 12 20
Dir  ;display the current directory
Screen Save  ;save the screen
Clear Screen  ;clear the screen
Screen Restore          ;restore the screen
```

See Also: Clear Screen, Screen Save, Type File, Type Line, Type String

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Screen Save

Description: This command applies to the remote PC's display buffer. Use it to take a snapshot of the screen. Everything on the screen, including the cursor type and position, is stored to a buffer so that it can be restored later in the script. To restore the screen, use Screen Restore. For example, when an error occurs that requires the user's attention, save the screen, display information that lets the user remedy the error, then restore the screen and continue processing.

Syntax: **Screen Save**

Reserved Variables: (none)

Example: The following example saves the contents of the screen and later restores those contents.

```
;move the cursor to row 12, column 20
Cursor Position 12 20
Dir  ;display the current directory
Screen Save  ;save the screen
Clear Screen  ;clear the screen
Screen Restore          ;restore the screen
```

See Also: Clear Screen, Screen Restore, Type Line, Type File, Type String

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Script

Description: Use this command to launch a second scriptfrom within the current script. The first script is the calling script, and the second script is the called script

Control returns to the first script when the second script terminates (with an End command), and the first script's execution continues at the line immediately following the one containing the Script command. Scripts can be nested ten levels deep.

Syntax: **Script** script_name

Parameters:

script_name	String literal or string variable containing the name of the script to execute. Do not include the .SCR filename extension.

Reserved Variables:

$Result	Set to the integer value (if any) returned from the second script when it ends.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: In the following example, MAIN.SCR calls ANOTHER.SCR. After ANOTHER.SCR completes, MAIN.SCR continues executing.

```
MAIN.SCR
Integer $1, $2 ;declare $1 and $2 as global integers
$1 = 5
$2 = 5
Type Line "The value is: "
Script ANOTHER          ;call ANOTHER.SCR script file
;print product of $1 and $2 (25) to the screen
Type Line $Result
End Terminal  ;end script processing

ANOTHER.SCR
value = $1 * $2          ;calculate product of $1 and $2
;end script processing, return control
;and pass value to MAIN.SCR
End value
```

See Also: GoSub, GoTo, Link, Return

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Seek

Description: This command applies to the remote PC only. Use it to position the file pointer at a specified location in the file. The next Read or Write command starts at that position. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Seek** file_number offset option

Parameters:

file_number — Integer literal used by the Open or Create command to identify the file. Valid identifiers are 1 through 9.

offset — Integer literal or integer variable containing the number of spaces to move, or offset, from the starting point specified by the option parameter. A positive number moves the pointer to the right; a negative number moves the pointer to the left.

option — Integer literal or integer variable declaring the starting point for the offset parameter.

Valid values for option are

:

Value	**Result**
0	Moves the pointer to the beginning of the file and then moves the pointer the value of offset. (Offset must be positive.)
1	Moves the pointer from the current position the value of offset.
2	Moves the pointer to the end of the file plus the value of offset. (Offset must be negative.)

Reserved Variables:

$Result — Set to the new position of the pointer within the file. If an error occurs, set to the value of $Error. If you move the pointer past either end of the file, the end-of-file error (-12) occurs.

$Error — Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example opens a sharable read/write file (NEWFILE.TXT), moves the read/write pointer to the end, appends a string, then closes the file.

```
Open 1 "newfile.txt" "SW"
Seek 1 0 2
Write String 1 "End of File."
Close 1
```

See Also: Close, Open, Read Line, Read String, Write Line, Write String

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Send Char

Description:	Use this command to send a single character using the currently active communications device. You can send characters as strings or their ASCII decimal value.	
Syntax:	**Send Char** character	
Parameters:	character	A literal or variable (that can be either an integer or string type) containing the character to send. If a string is specified, only the first character of the string is sent.
Reserved Variables:	$Result	Set to a non-negative integer value. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example sends a carriage return followed by an A.

```
Send Char 13 ;send a carriage return
Send Char "A"        ;send the character "A"
```

See Also: Receive Char, Receive Clear, Receive File, Receive Line, Receive String, Send Clear, Send File, Send Line, Send String

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Send Clear

Description: Use the command to clear the send buffer of any data not yet transmitted. Data can be in the send buffer if the communications rate is slow (such as a low serial data rate) or if the transmission is halted due to a flow control error or recovery situation.

Syntax: **Send Clear**

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example clears the send buffer before it sends a string.

```
String string1[14]
string1 = "Initializing..."
Wait Silence 5          ;wait for 5 seconds of line silence
Send Clear     ;clear the send buffer
Send String string1
```

See Also: Receive Char, Receive Clear, Receive File, Receive Line, Receive String, Send Char, Send File, Send Line, Send String

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Send File

Description:	Use this command to send a file using the protocol specified by the Set Protocol command. When the transfer begins, a status screen displays the transfer's progress.	
Syntax:	**Send File** filename	
Parameters:	filename	String literal or string variable containing the full name and path of the outgoing file. For multi-file protocols, the name may contain wildcards (? or *).
Reserved Variables:	$Result	Set to a non-negative integer value. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, "Error Messages."
Example:	The following example sends a file using the ZMODEM protocol. Set Protocol zmodem Send File "c:\data\datafile.txt"	
See Also:	Receive Char, Receive Clear, Receive File, Receive Line, Receive String, Send Char, Send Clear, Send Line, Send String	

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Send Line

Description:	Use this command to send a string and its terminating character through the communications port. The terminating character is the value of the reserved variable $SLineEnd (whose default value is a carriage return).	
Syntax:	**Send Line** string	
Parameters:	string	String literal or string variable containing the data to send.
Reserved Variables:	$Result	Set to the number of characters sent. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example sends a string followed by the line-terminating character(s) in $SLineEnd.

```
String string1[14]
string1 = "Processing..."
Send Line string1        ;send string1
```

See Also: Receive Char, Receive Clear, Receive File, Receive Line, Receive String, Send Char, Send Clear, Send File, Send String

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Send ScanCode

M scan code, then send them through the current communications port.

Syntax: **Send ScanCode** string [length]

Parameters:

string	String literal or string variable containing the data to send.	
length	(Optional) integer containing the number of characters to send. The default value is the size of string.	

Reserved Variables:

$Result	Set to the number of characters sent. If an error occurs, set to the value of $Error.	
$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’	

Example: The following example sends the codes for D, I, R, Ctrl, and M. Ctrl+M is a carriage return.

```
Send ScanCode "Dir^M"
```

The next example sends codes for W, e, l, c, o, m, e and !.

```
String string1[14]
string1 = "Welcome!"
Send ScanCode string1
```

See Also: Dial Host

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Send String

Description:	Use this command to send a string through the communications port. The entire string is sent, unless a length is specified. No terminating characters are sent.	
Syntax:	**Send String** string [length]	
Parameters:	string	String literal or string variable containing the data to send.
	length	(Optional) integer literal or integer variable containing the number of characters to send. The default value is the size of string.
Reserved Variables:	$Result	Set to the number of characters sent. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, "Error Messages."
Example:	The following example sends a password.	
	String password[14]	
	password = "parley"	
	Send String password	
See Also:	Receive Char, Receive Clear, Receive File, Receive Line, Receive String, Send Char, Send Clear, Send File, Send Line	

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Session Delay

Description: Use this command to specify the number of minutes between retries of the Session Dial command.

Syntax: **Session Delay** [minutes]

Parameters:

minutes	(Optional) integer literal or integer variable containing the number of minutes to delay. The default is 0.

Reserved Variables: (none)

Example: The following example calls the Office PC and sends a file remotely. The Session Delay command stipulates the number of minutes to wait before redialing.

```
Session ExitMode Accept          ;mode to leave host in
Session OnError End     ;what to do if error occurs
;what to do if file conflict occurs
Session Overwrites Older
;set the timeout value in seconds
Session Timeout 10
;set the number of retries to attempt
Session Retry 3
;set the amount of time between retries in minutes
Session Delay 30
;dial Office PC at 4 A.M. on May 4, 1993
Session Dial "Office PC" 0400 940504
;send a file from the remote to the host
SessOpr Remote Send "c:\data\wk1030.txt"
Session End   ;terminate the session
```

See Also: Session Dial, Session End, Session ExitMode, Session Retry, Session Overwrites, Session Timeout, SessOpr Host Run, SessOpr Host Send, SessOpr Remote Run, SessOpr Remote Send

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Session Dial

Description: Use this command to initiate a remote control session by dialing a pcAnywhere host PC.

Syntax: **Session Dial** host_name [time [date]] [script_name]

Parameters:

host_name	String literal or string variable containing the name of the remote control session configuration.
time	(Optional) integer literal or integer variable containing the time of day that you want to first attempt the connect. The time format is HHMM. The default is the current time.
date	(Optional) integer literal or integer variable containing the date to first attempt the connect. The date parameter cannot be used unless a value for time is also specified. Its format is YYMMDD. The default is the current date.
script_name	(Optional) string literal or string variable containing the name of the script file to launch after connection is made. Do not include the .SCR file extension.

Reserved Variables:

$Result	Set to a non-negative integer value. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example calls the Office PC and sends a file remotely.

```
Session ExitMode Accept          ;mode to leave host in
Session OnError End    ;what to do if error occurs
;what to do if file conflict occurs
Session Overwrites Older
;set the timeout value in seconds
Session Timeout 10
;set the number of retries to attempt
Session Retry 3
;set the amount of time between retries in minutes
Session Delay 30
;dial Office PC at 4 A.M. on May 4, 1993
Session Dial "Office PC" 0400 940504
;send a file from the remote to the host
SessOpr Remote Send "c:\data\wk1030.txt"
Session End  ;terminate the session
```

See Also: Session Delay, Session End, Session Exit Mode, Session OnError, Session Overwrites, Session Retry, Session Timeout, SessOpr Host Run, SessOpr Host Send, SessOpr Remote Run, SessOpr Remote Send

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Session End

Description: Use this command to terminate a session initiated with a Session Dial command. If the phone is still connected when the Session End command is executed, the phone is hung up

Syntax: **Session End**

Reserved Variables: (none)

Example: The following example calls the Office PC and sends a file remotely. Then it terminates the session.

```
Session ExitMode Accept              ;mode to leave host in
Session OnError End     ;what to do if error occurs
;what to do if file conflict occurs
Session Overwrites Older
;set the timeout value in seconds
Session Timeout 10
;set the number of retries to attempt
Session Retry 3
;set the amount of time between retries in minutes
Session Delay 30
;dial Office PC at 4 A.M. on May 4, 1993
Session Dial "Office PC" 0400 940504
;send a file from the remote to the host
SessOpr Remote Send "c:\data\wk1030.txt"
Session End   ;terminate the session
...
```

See Also: Session Delay, Session Dial, Session ExitMode, Session OnError, Session Overwrites, Session Retry, Session Timeout , SessOpr Host Run, SessOpr Host Send, SessOpr Remote Run, SessOpr Remote Send

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Session ExitMode

Description: Use this command to specify the state in which to leave the host PC when a remote control session terminates. This command must be executed before the Session Dial command. If the host PC is configured to prevent the remote user from modifying the host state, this command has no effect.

Syntax: **Session ExitMode Wait** | **Idle** | **Accept** | **Original** | **Cancel**

Parameters:

Wait Disconnects remote user from the host, leaving the pcAnywhere host as the active application. No host activities can be performed until there is a connection. See note below.

Idle Disconnects remote user from the host, leaving pcAnywhere running in the background. A session cannot begin until a host user presses the host hotkey. See note below.

Accept Disconnects remote user from the host, leaving the host to automatically accept another call. pcAnywhere stays in memory and a session starts when a remote user calls the host—no action is required from the host user. In fact, the host user cannot prevent calls from coming in.

Also use this mode when you want to start a possibly long job on the host, hang up, then call back later to check the results. The host PC can continue processing even though the remote PC is no longer watching.

Original Disconnects remote user from the host and leaves the host in the same state it was in before the remote session occurred.

Cancel Disconnects remote user from the host and removes the host software from memory, preventing the host from accepting incoming calls.

NOTE: In pcAnywhere for Windows, when the host TSR is not running, Idle and Wait have the same functionality as Accept.

Reserved Variables: $SesHostExitMode Set to 0 for Wait, 1 for Idle, 2 for Accept, 3 for Original, or 4 for Cancel.

Example: The following example sets the exit mode to accept, calls the Office PC, and sends a file remotely.

```
Session ExitMode Accept          ;mode to leave host in
Session OnError End    ;what to do if error occurs
;what to do if file conflict occurs
Session Overwrites Older
;set the timeout value in seconds
Session Timeout 10
;set the number of retries to attempt
Session Retry 3
;set the amount of time between retries in minutes
Session Delay 30
;dial Office PC at 4 A.M. on May 4, 1993
Session Dial "Office PC" 0400 940504
;send a file from the remote to the host
```

```
SessOpr Remote Send "c:\data\wk1030.txt"
Session End   ;terminate the session
```

See Also: Session Delay, Session Dial, Session End, Session OnError, Session Overwrites, Session Retry, Session Timeout, SessOpr Host Run, SessOpr Host Send, SessOpr Remote Run, SessOpr Remote Send

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Session OnError

Description: Use this command to specify what should occur if one of the following commands fails: SessOpr Host Run, SessOpr Host Send, SessOpr Remote Run, or SessOpr Remote Send. The command can terminate the script or allow processing to continue. This command takes precedence over the On Error command.

The Session OnError command does not reset $Error to zero before the next command is executed. You should do that to avoid reprocessing one error over and over.

Syntax: **Session OnError Ignore** | **Next** | **End**

Parameters:

Ignore	Ignores the command that caused the error and advances to the next command.
Next	Advances to the matching Session End command.
End	Ends the script and returns control to pcAnywhere.

Reserved Variables:

$SesOnError	Set to 0 for Ignore, 1 for Next, or 2 for End.

Example: The following example specifies that the script should end if an error occurs. Then it calls the Office PC and sends a file remotely.

```
Session ExitMode Accept          ;mode to leave host in
Session OnError End     ;what to do if error occurs
;what to do if file conflict occurs
Session Overwrites Older
;set the timeout value in seconds
Session Timeout 10
;set the number of retries to attempt
Session Retry 3
;set the amount of time between retries in minutes
Session Delay 30
;dial Office PC at 4 A.M. on May 4, 1993
Session Dial "Office PC" 0400 940504
;send a file from the remote to the host
SessOpr Remote Send "c:\data\wk1030.txt"
Session End   ;terminate the session
```

See Also: Session Delay, Session Dial, Session End, Session ExitMode, Session Overwrites, Session Retry, Session Timeout, SessOpr Host Run, SessOpr Host Send, SessOpr Remote Run, SessOpr Remote Send

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Session OverWrites

Description: Use this command to specify whether or not to overwrite existing files during the execution of a SessOpr Host Send or SessOpr Remote Send command. Execute this command before attempting file transfers, so script execution is not suspended when a conflict occurs.

Syntax: **Session OverWrites Always** | **Never** | **Older**

Parameters:

Always Causes the destination file to be overwritten.

Never Causes the file sent to be ignored.

Older Causes the file at the destination to be overwritten if the file being transferred is more recent (compares date/time stamp).

Reserved Variables: $SesOverwrites Set to 0 for Always, 1 for Never, or 2 for Older.

Example: The following example specifies that transferred files can overwrite files that are older. Then it calls the Office PC and sends a file remotely.

```
Session ExitMode Accept          ;mode to leave host in
Session OnError End    ;what to do if error occurs
;what to do if file conflict occurs
Session OverWrites Older
;set the timeout value in seconds
Session Timeout 10
;set the number of retries to attempt
Session Retry 3
;set the amount of time between retries in minutes
Session Delay 30
;dial Office PC at 4 A.M. on May 4, 1993
Session Dial "Office PC" 0400 940504
;send a file from the remote to the host
SessOpr Remote Send "c:\data\wk1030.txt"
Session End  ;terminate the session
```

See Also: Session Delay, Session Dial, Session End, Session ExitMode, Session OnError, Session Retry, Session Timeout, SessOpr Host Run, SessOpr Host Send, SessOpr Remote Run, SessOpr Remote Send

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Session Retry

Description: Use this command to specify the number of times to redial the host specified in the Session Dial command.

Syntax: **Session Retry** [count]

Parameters:

count	(Optional) integer literal or integer variable containing the number of times to attempt initiating a session. The default is 1.

Reserved Variables: (none)

Example: The following example sets the number of retries to three. Then it calls the Office PC and sends a file remotely.

```
Session ExitMode Accept          ;mode to leave host in
Session OnError End     ;what to do if error occurs
;what to do if file conflict occurs
Session Overwrites Older
;set the timeout value in seconds
Session Timeout 10
;set the number of retries to attempt
Session Retry 3
;set the amount of time between retries in minutes
Session Delay 30
;dial Office PC at 4 A.M. on May 4, 1993
Session Dial "Office PC" 0400 940504
;send a file from the remote to the host
SessOpr Remote Send "c:\data\wk1030.txt"
Session End   ;terminate the session
```

See Also: Session Delay, Session Dial, Session End, Session ExitMode, Session Overwrites, Session Timeout, SessOpr Host Run, SessOpr Host Send, SessOpr Remote Run, SessOpr Remote Send

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Session Timeout

Description: Use this command to specify the inactivity timeout value (in seconds) for a remote control session. Use for DOS sessions only.

Syntax: **Session Timeout** seconds

Parameters:

seconds	Integer literal or integer variable specifying the number of seconds before a timeout occurs.

Reserved Variables:

$SesTimeout	Set to specified number of seconds.

Example: The following example specifies a timeout after 10 seconds, then calls the Office PC and sends a file remotely.

```
Session ExitMode Accept          ;mode to leave host in
Session OnError End     ;what to do if error occurs
;what to do if file conflict occurs
Session Overwrites Older
;set the timeout value in seconds
Session Timeout 10
;set the number of retries to attempt
Session Retry 3
;set the amount of time between retries in minutes
Session Delay 30
;dial Office PC at 4 A.M. on May 4, 1993
Session Dial "Office PC" 0400 940504
;execute a file on the host
SessOpr Host Run "c:\dos\edit.com"
;edit.com ends when the user exits or when session timeout seconds expires
SessOpr Remote Send “wk1030.txt” “mrd.txt”
Session End   ;terminate the session
```

See Also: Session Delay, Session Dial, Session End, Session ExitMode, Session OnError, Session Overwrites, Session Retry, SessOpr Host Run, SessOpr Host Send, SessOpr Remote Run, SessOpr Remote Send

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

SessOpr Host Run

Description: Use this command to launch an application on the host PC during a session initiated with a Session Dial command. If you do not specify Wait or NoWait, Wait is the default.

When using this command to run DOS commands from COMMAND.COM, such as COPY, you must use COMMAND.COM with the /C option as part of the command line:

```
SessOpr Host Run "command /c copy *.* b:"
```

Syntax: **SessOpr Host Run** [**Wait** | **NoWait**] command_line

Parameters:

Wait	(Optional) causes the script to pause until the launched program is completed. The launched program ends when the user exits it or it is automatically terminated after the number of seconds specified by the Session Timeout command.
NoWait	(Optional) tries to launch the specified program and continue. Used under Windows only.
command_line	String literal or string variable containing the command line for the application.

Reserved Variables:

$Result	Set to the errorlevel returned by the application.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example calls the Office PC and remotely executes a file on the host.

```
Session ExitMode Accept          ;mode to leave host in
Session OnError End    ;what to do if error occurs
;what to do if file conflict occurs
Session Overwrites Older
;set the timeout value in seconds
Session Timeout 10
;set the number of retries to attempt
Session Retry 3
;set the amount of time between retries in minutes
Session Delay 30
;dial Office PC at 4 A.M. on May 4, 1993
Session Dial "Office PC" 0400 940504
;execute file on the host
SessOpr Host Run "c:\dos\move.bat"
Session End  ;terminate the session
```

See Also: Session Delay, Session Dial, Session End, Session ExitMode, Session OnError, Session Overwrites, Session Retry, Session Timeout, SessOpr Host Send, SessOpr Remote Run, SessOpr Remote Send

Script Editor Commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

SessOpr Host Send

Description: Use this command to send a file from the host PC to the remote PC during a session initiated with a Session Dial command. It automates file transfers when connected to a pcAnywhere host. Be sure to use the Session Overwrites command to specify what to do if the file you are transferring already exists at the destination.

Syntax: **SessOpr Host Send** source_dest

Parameters:

source_dest — String literal or string variable containing:

a (Required) the complete or partial pathname to the file(s) being sent. Wildcards (* or ?) are permitted.

b (Optional) a space followed by the complete or partial destination pathname. If one file is sent, this can be a filename. Must be encased in separate set of quotes from source file and pathname.

The current directory is always the default path.

Reserved Variables:

$Result — Set to the number of files actually sent. If an error occurs, set to the value of $Error.

$Error — Standard errors. See Appendix B, "Error Messages."

Example: The following example calls the Office PC and copies a file from the host.

```
Session ExitMode Accept          ;mode to leave host in
Session OnError End    ;what to do if error occurs
Session Overwrites Older
Session Timeout 10
Session Retry 3
Session Delay 30
;dial Office PC at 4 A.M. on May 4, 1993
Session Dial "Office PC" 0400 940504
;send a file from the host to the remote
SessOpr Host Send "\budget\*.xls" "\updates"
Session End   ;terminate the session
```

See Also: Session Delay, Session Dial, Session End, Session ExitMode, Session OnError, Session Overwrites, Session Retry, Session Timeout, SessOpr Host Run, SessOpr Remote Run, SessOpr Remote Send

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

SessOpr Remote Run

Description: Use this command to launch an application on the remote PC during a session initiated with a Session Dial command. If you do not specify Wait or NoWait, Wait is the default.

When using this command to run DOS commands from COMMAND.COM, such as COPY, you must use COMMAND.COM with the /C option as part of the command line:

```
SessOpr Remote Run "command.com/c copy *.* b:"
```

Syntax: **SessOpr Remote Run** [**Wait** | **NoWait**] command_line

Parameters:

Wait	(Optional) causes the script to pause until the launched program is completed. The launched program ends when the user exits it or it is automatically terminated after the number of seconds specified by the Session Timeout command.
NoWait	(Optional) tries to launch the specified program and continue. Used under Windows only.
command_line	String literal or string variable containing the command line for the application.

Reserved Variables:

$Result	Set to the errorlevel returned by the application. If an error occurs, set to the value $Error.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example calls the Office PC then executes a file from the remote.

```
Session ExitMode Accept          ;mode to leave host in
Session OnError End    ;what to do if error occurs
Session Overwrites Older
Session Timeout 10
Session Retry 3
;set the amount of time between retries in minutes
Session Delay 30
;dial Office PC at 4 A.M. on May 4, 1993
Session Dial "Office PC" 0400 940504
SessOpr Remote Run "c:\dos\copy.bat"
Session End  ;terminate the session
```

See Also: Session Delay, Session Dial, Session End, Session ExitMode, Session OnError, Session Overwrites, Session Retry, Session Timeout, SessOpr Host Run, SessOpr Host Send, SessOpr Remote Send

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

SessOpr Remote Send

Description: Use this to send a file from the remote PC to the host PC during a session initiated with a Session Dial command. It automates file transfers when connected to a pcAnywhere host. Be sure to use the Session Overwrites command to specify what to do if the file you are transferring already exists at the destination.

Syntax: **SessOpr Remote Send** source_dest

Parameters:

source_dest — String literal or string variable containing:

- **a** (Required) the complete or partial pathname to the file(s) being sent. Wildcards (* or ?) are permitted.
- **b** (Optional) a space followed by the complete or partial destination pathname. If one file is sent, this can be a filename.

The current directory is always the default path.

Reserved Variables:

$Result — Set to the number of files actually sent. If an error occurs, set to the value of $Error.

$Error — Standard errors. See Appendix B, "Error Messages."

Example: The following example calls the Office PC and sends a file remotely.

```
Session ExitMode Accept          ;mode to leave host in
Session OnError End    ;what to do if error occurs
;what to do if file conflict occurs
Session Overwrites Older
;set the timeout value in seconds
Session Timeout 10
;set the number of retries to attempt
Session Retry 3
;set the amount of time between retries in minutes
Session Delay 30
;dial Office PC at 4 A.M. on May 4, 1993
Session Dial "Office PC" 0400 940504
;send a file from the remote to the host
SessOpr Remote Send "wk1030.txt" "mrd.txt"
Session End  ;terminate the session
```

See Also: Session Delay, Session Dial, Session End, Session ExitMode, Session OnError, Session Overwrites, Session Retry, Session Timeout, SessOpr Host Run, SessOpr Host Send, SessOpr Remote Run

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set Attribute

Description: This command applies to the remote PC's display buffer. Use it to set colors and other screen attributesfor subsequent characters output to the screen. You can set only one attribute at a time, though you can have more than one setting active. All changes effect the characters and spaces following the current cursor position (not inclusive). Settings 1 through 7 are cumulative. Settings 22 through 27 are used to selectively disable settings 1 through 7. Selecting setting 0 disables all settings 1 through 7.

You can have only one foreground and one background color active at any time. For example, you can set the screen attributes to a blinking blue foreground on a black background.

Syntax: **Set Attribute** attribute

Parameters: attribute Integer literal or integer variable containing the attribute setting.

Valid settings are:

Value	**Video Attribute**
0	Normal (disables settings 1 through 7)
1	Intense
4	Underline
5	Blink
7	Reverse
22	Intense Off
24	Underline Off
25	Blink Off
27	Reverse Off

Value	**Foreground Colors**
30	Black
31	Red
32	Green
33	Brown
34	Blue
35	Magenta

36	Cyan
37	Light Gray
Value	**Background Colors**
40	Black
41	Red
42	Green
43	Brown
44	Blue
45	Magenta
46	Cyan
47	Light Gray

Reserved Variables: (none)

Example: The following examples set and then display a variety of attributes.

```
Set Attribute 30        ;set black foreground
Set Attribute 41        ;set red background
Type String "This is displayed black on red.^M^J"
Set Attribute 44        ;set blue background
Type String "And this is displayed black on blue.^M^J"
Set Attribute 5;set the text to blink
Type String "And this is displayed blinking black on blue."
Set Attribute 0;turn off the blinking
Type String "Now the blinking is turned off."
```

See Also: Clear Screen, Set Palette, Set Quiet

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set Cancel

Description: Use this command to specify what happens when the user presses Esc. When on, Wait and Receive commands are terminated and, if an On Cancel command is active, the command specified by On Cancel is executed.

Syntax: **Set Cancel On** | **Off**

Parameters:

On	Causes Esc to be understood as canceling.	
Off	Causes Esc to be ignored.	

Reserved Variables:

$Result	Set to 0.	
$Error	Unchanged.	
$Cancel	Set to 1 for on or 0 for off.	

Example: The following example enables canceling and specifies what is to happen if the user presses Esc.

```
Set Cancel On          ;enable canceling
On Cancel GoTo @end
;attempt to connect with CompuServe
Dial OnlSvc "CompuServe"
Wait Time 2
;send carriage return to alert host
Send Char 13
;end script processing and return to terminal mode
End Terminal


@end:
End;end script processing
```

See Also: On Cancel, Set Disconnect, Set Timeout, Wait and Receive commands

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set CharDelay

Description:	Use this command to set the default value of the between-character send delay. The Send String and Send Line commands insert this delay between each character sent. For most applications, delays are not necessary, though some mainframes require that data be entered at a "typing" speed. The current value of the between-character send delay is stored in the $CDelay reserved variable.	
Syntax:	**Set CharDelay** delay	
Parameters:	delay	Integer literal or integer variable specifying the delay value (in tenths of a second).
Reserved Variables:	$Result	Set to 0.
	$Error	Unchanged.
	$CDelay	Set to the specified delay.

Example: The following example sets the character delay to 1/10th of a second.

```
Set CharDelay 1
String string1[14]
string1 = "Processing..."
Send String string1
```

See Also: Send Line, Send String

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set Disconnect

Description: Use this command to specify what happens when a connection is lost. When on, Wait and Receive commands are terminated, and if an On Disconnect command is active, the command specified by On Disconnect is executed.

Syntax: **Set Disconnect On** | **Off**

Parameters:

On	Causes a lost connection to end Wait and Receive commands.
Off	Causes no action to be taken when a connection is lost.

Reserved Variables:

$Result	Set to 0.
$Error	Unchanged.
$Disconnect	Set to 1 for on or 0 for off.

Example: If the connection is lost during the execution of the following example, the script ends.

```
Set Disconnect On       ;enable disconnect action
On Disconnect GoTo @end
;attempt to connect with CompuServe
Dial OnlSvc "CompuServe"
Wait Time 2
Send Char 13
;end script processing and return to terminal mode
End Terminal

@end:
End;end script processing
```

See Also: On Disconnect, Set Cancel, Set Timeout, Wait and Receive commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set DTR

Description: This command applies only to serial communications. Use it to send the RS-232 Data Terminal Ready (DTR) signal to the serial port. When on, the signal is raised or ready. When off, the signal is lowered or not ready. If you load a hardware configuration after this command, the state of the signal is reset to that in the configuration.

Syntax: **Set DTR On** | **Off**

Parameters:

On	Raises the DTR signal.
Off	Lowers the DTR signal.

Reserved Variables:

$DTR	Set to 1 for on or 0 for off.

Example: The following example lowers and then raises the DTR signal.

```
Set DTR Off   ;lower the DTR signal
Wait 1        ;for one second
Set DTR On    ;then raise it again
```

See Also: Exit Online

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set Echo

Description:	Use this command to enable or disable the echoing received during Wait and Receive commands. Set Echo must be on to display characters received while in the terminal mode. By default, Set Echo is set to off. Therefore, you usually turn it on during the log-on process to display the log-on prompt on the remote.	
Syntax:	**Set Echo On** | **Off**	
Parameters:	**On**	Allows the display of received characters.
	Off	Stops the display of received characters.
Reserved Variables:	$Echo	Set to 1 for on or 0 for off.

Example: The following example specifies the display of responses, dials the host, and waits for a string.

```
Set Echo On  ;on to display responses
Dial Host "Office PC"  ;dial the host
;wait for and display password prompt
Wait String "Enter password:"
```

See Also: Set Cancel, Set Disconnect

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set File Attr

Description: This command applies to the remote PC only. Use it to set the attributes of the specified file. The file must be closed before performing this operation. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Set File Attr** filename new_attribute

Parameters:

filename	String literal or string variable containing the name of the file to modify.
new_attribute	Integer literal or integer variable containing the sum of the values for the desired attributes. The values are as follows:

Attribute	**Value**
Read-only	1
Hidden	2
System file	4
Volume label	8
Subdirectory	16
Archive	32

Using 0 as the attribute, makes the file a Normal file.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, “Error Messages.”

Example: The following example makes DATA.FIL a read-only file.

```
Set File Attr "C:\DATA.FIL" 1
```

See Also: Get File Attr, Get File Date, Get File Size, Get File Time, Set File Date, Set File Size, Set File Time

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set File Date

Description: This command applies to the remote PC only. Use it to revise the date of a file which has otherwise not changed. The file must be closed before performing this operation. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

MS-DOS changes the date of a file when a file is closed after being opened. In other words, the date is changed when a file is updated, but not when it is renamed, copied, moved, or has changed its file attributes.

Syntax: **Set File Date** filename new_date

Parameters:

filename	String literal or string variable containing the name of the file.	
new_date	A literal or variable (that can be either an integer or string type) containing the new date in the format YYYYMMDD.	

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example sets the date of WK1030.DAT to 11/10/1999.

```
Set File Date "wk1030.dat" 19991110
```

See Also: Get File Attr, Get File Date, Get File Size, Get File Time, Set File Attr, Set File Size, Set File Time

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set File Size

Description: This command applies to the remote PC only. Use it to change the indicated size of the specified file. The file must be closed before performing this operation. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

WARNING: Use this command cautiously. Changing the way DOS indicates the size of a file may cause unintentional errors that are extremely difficult to detect.

Syntax: **Set File Size** filename new_size

Parameters:

filename	String literal or string variable containing the name of the file.
new_size	A literal or variable (that can be either an integer or string type) containing the new file size (in bytes).

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example changes the size of TEMP.DAT to 1024 bytes.

```
Set File Size "temp.dat" 1024
```

See Also: Get File Attr, Get File Date, Get File Size, Get File Time, Set File Attr, Set File Date, Set File Time

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set File Time

Description: This command applies to the remote PC only. Use it to revise the time of a file which has otherwise not changed. The file must be closed before performing this operation. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

MS-DOS changes the time of a file when a file is closed after being opened. In other words, the time is changed when a file is updated, but not when it is renamed, copied, moved, or has changed its file attributes.

Syntax: **Set File Time** filename new_time

Parameters:

filename	String literal or string variable containing the name of the file.	
new_time	A literal or variable (that can be either an integer or string type) containing the new time in the format HHMM.	

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example changes the time of NEWFILE.DAT to 8 A.M.

```
Set File Time "newfile.dat" 0800
```

See Also: Get File Attr, Get File Date, Get File Size, Get File Time, Set File Attr, Set File Date, Set File Size

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set Flow

Description: This command is generally used only for serial connections. Use it to set flow control. You can use this command before or after making the connection. However, if you load a hardware configuration after this command, the flow is reset to that in the configuration.

XON/XOFF flow control is commonly known as software flow control, and RTS/CTS is known as hardware flow control. RTS/CTS flow control is achieved by varying the voltage on the RTS/CTS signal lines between the terminal and the modem. XON/XOFF flow control is achieved by embedding control characters in the data stream.

RTS/CTS flow control is preferred. Use XON/XOFF flow control when RTS/CTS flow control is not supported, or when otherwise required.

Syntax: **Set Flow RTSCTS** | **XONXOFF** | **Both** | **None**

Parameters:

RTSCTS	Sets hardware flow control on and software control off.
XONXOFF	Sets software flow control on and hardware control off.
Both	Sets hardware and software flow control on.
None	Sets hardware and software flow control off.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, “Error Messages.”

Example: The following example sets flow control to RTS/CTS.

```
Set Flow RTSCTS
```

See Also: Set RTS

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set IgnoreCase

Description: Use this command to turn case-checking off or on when comparing strings or waiting for strings. Set it to off to differentiate between uppercase and lowercase letters. The default for Set IgnoreCase is on, which means string comparisons are not case sensitive This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Set IgnoreCase On** | **Off**

Parameters:

On	Ignores case.
Off	Checks case.

Reserved Variables:

$Result	Set to 0.
$Error	Unchanged.
$IgnoreCase	Set to 1 for on or 0 for off.

Example: The following example ignores the case when comparing "ABC" and "abc".

```
;case will be ignored by default
StrCmp "ABC" "abc"      ;$Result is set to 0
```

The next example uses case when comparing "ABC" and "abc".

```
Set IgnoreCase Off      ;case is not ignored
StrCmp "ABC" "abc"      ;$Result is set to -1
```

See Also: Lower, StrCmp, Upper, Wait String

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set Palette

Description: This command is valid in Norton pcAnywhere for DOS scripts only. It applies to the remote PC's display buffer. Each displayable object, such as a menu or form, has fields that are predefined as having normal, intense, or reverse video attributes. For example, the options listed on a menu are normal; the selected option is reverse, and both the menu's title and border are intense. Use this command to set a foreground and background color combination for the specified video attribute on the specified displayable object. Values set in a script file remain in effect until the user exits pcAnywhere.

NOTE: If you are using pcAnywhere on a monochrome monitor, try to use colors that have high contrast. Darker shades are more likely to be distinguishable.

Syntax: **Set Palette** attribute color

Parameters: attribute — Integer literal or integer variable indicating both the displayable object and the video attribute to change.

	Normal Video	**Intense**	**Reverse Video**
Menus	3	4	5
Yes/No menus	6	7	8
Errors/Warnings	12	13	14
Help	15	16	17
Status/Info	18	19	20
Terminal Mode	24	25	26
Forms	27	28	29

color — Hexadecimal integer starting with 0x and indicating the combination of foreground and background colors. Of the last two digits, the background color is specified before the foreground color.

black	0
blue	1
brown	6
cyan	3
dark gray*	8
green	2
light blue*	9

light cyan*	B
light green*	A
light magenta*	D
light red*	C
light gray	7
magenta	5
red	4
white*	F
yellow*	E

*Can be used as foreground colors only

Reserved Variables: (none)

Example: The following example sets terminal mode to normal and sets the color to blue on light gray.

```
Set Palette 24 0x71
```

See Also: Screen Restore, Screen Save, Set Attribute, Set Echo, Set Quiet

Script Editor Commands

Set Parity

Description: This command applies to serial connections only. Use it to change the current parity type. The parity determines the settings for data bits and stop bits (see the table below). Both computers must use the same parity setting. You can use this command before or after making the connection. However, if you load a hardware configuration after this command, the parity is reset to that in the configuration.

Parity	Data Bits	Parity Bit	Stop Bit
None	8	N	1
Even	7	E	1
Odd	7	O	1
Mark	7	M	1
Space	7	S	1

Syntax: **Set Parity None** | **Even** | **Odd** | **Mark** | **Space**

Parameters:

None Use this setting for direct connections and for modem communications where both the host and remote PCs are running pcAnywhere.

Even, Odd, Mark, Space Use one of these settings when communicating with a non-pcAnywhere host PC that requires a specific parity setting.

Reserved Variables: (none)

Example: The following example specifies even parity.

```
Set Parity Even
```

See Also: Dial Host, Dial OnlSvc, Load HostInfo, Load OnlSvcInfo

Script Editor Commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set Port

Description:	Use this command to select the communications portfor subsequent Set Speed, Set Parity, or Set Flow commands. It changes the port from the one specified by the current configuration. Use the Set Port command prior to making a connection. If you load a hardware configuration after this command, the port is reset to that in the configuration.	
Syntax:	**Set Port** number	
Parameters:	number	Integer representing communications port to be affected. Valid values are 1 through 4.
Reserved Variables:	$Result	Set to 0.
	$Error	Set if an invalid port value is specified.

Example: This example combines the Set Port command with the Set Parity command for a special situation that requires a temporary adjustment of parity.

```
Set Port 1      ;selects com 1
Set Speed 9600          ;sets bps rate
Dial Number "1-408-973-9834"      ;dials service
Set Parity Even         ;changes parity to even
```

See Also: Dial Host, Set Flow, Set Parity, Set Speed

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set Protocol

Description: Use this command to set the file-transfer protocol to be used for subsequent Send File and Receive File commands. You can use this command before or after making the connection. However, if you load a session configuration after this command, the protocol is reset to that in the configuration.

ZMODEM is generally regarded to be the fastest file-transfer protocol, although pcAnywhere's proprietary pcAnywhere protocol (available only when using AWSEND.EXE) has been tested to be faster. The pcAnywhere protocol uses a CRC algorithm for error-correcting and verifying the integrity of received files, as does ZMODEM. Both protocols have crash-recovery ability, which enables a partially completed file transfer to be resumed from the point of interruption. YMODEM-G is regarded by many to be faster than ZMODEM, although it has no inherent error-correcting ability, and should be used only with error-correcting modems, such as those employing MNP-5 or v.42bis error correction. YMODEM-G does not have crash recovery ability.

Syntax: **Set Protocol** type

Parameters:	type	Selected protocol. Valid types are XMODEM, XMODEM-CRC, XMODEM-1K, XMODEM-1KG, YMODEM, YMODEM-G, ZMODEM, KERMIT, ASCII.
Reserved Variables:	$Result	Set to 0.
	$Error	Unchanged.
	$Protocol	Set to an integer representing the specified protocol.

Example: The following example specifies that the ZMODEM protocol will be used when transferring files.

```
Set Protocol ZMODEM
```

See Also: Emulate, Receive File, Send File, Send String, Set Flow, Set RTS, Set Timeout

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set Quiet

Description:	This command applies to the remote PC's display buffer. Use it to display or hide progress messages about the commands in the script. The default state of Set Quiet is off, which displays the messages.	
Syntax:	**Set Quiet On** | **Off**	
Parameters:	**On**	Hides progress messages.
	Off	Displays progress messages.
Reserved Variables:	$Result	Set to 0.
	$Error	Unchanged.
	$Quiet	Set to 1 for on or 0 for off.
Example:	In the following example, information about the copy command is not displayed, although execution occurs. Set Quiet On COPY file1 file2	
See Also:	Set Echo	

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set Record

Description: Use this command to start and stop recording data. When set to on, all data received is written to the current recording file. The default recording file is AW.RCD. In the script, you can change the file by setting the reserved variable $RecordFile to a string containing another filename.

If you load a session configuration after this command, the value of $Record is reset to that in the configuration. So is the value of $RecordFile.

Syntax: **Set Record On** | **Off**

Parameters:

On	Records all received data in the current recording file.
Off	Stops recording of the current session.

Reserved Variables:

$Result	Set to 0.
$Error	Unchanged.
$Record	Set to 1 for on or 0 for off.

Example: The following example turns on recording, dials the host, then turns recording off.

```
Set Record On
Dial Host "Office PC"
...
Set Record Off
End
```

The following example turns on recording and dials the host. When the script ends, control passes to the remote user.

```
Set Record On
Dial Host "Office PC"
...
End Terminal
```

Set RTS

Description:	Use this command to set the state of the RS-232 Request to Send (RTS) signal. When on (or raised), pcAnywhere uses hardware flow control to direct the rate of data flow between your computer and modem. This is often used with high-speed and error-correcting modems. You can use this command before or after making the connection. However, if you load a hardware configuration after this command, the state of the signal is reset to that in the configuration.	
Syntax:	**Set RTS On** | **Off**	
Parameters:	**On**	Raises the RTS signal.
	Off	Lowers the RTS signal.
Reserved Variables:	$Result	Set to 0.
	$Error	Unchanged.
	$RTS	Set to 1 for on or 0 for off.
Example:	The following example lowers the RTS signal for one second, then raises it again.	
	Set RTS Off	
	Wait Time 1	
	Set RTS On	
See Also:	Set Flow	

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set Speed

Description: This command applies only to serial connections. It sets the bits-per-second (bps) speedof the communications port. The Set Speed command may be preceded by a Set Port command, which selects the communications port to be set. Use this command prior to making a connection. If you load a hardware configuration after this command, the speed is reset to that in the configuration.

The rate of the communications port affects the speed of data transmission between modems.

Syntax: **Set Speed** rate

Parameters:	rate	Integer literal or integer variable containing the bps rate. Permitted values are: 75, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600 or 115200.
Reserved Variables:	$Baud	Set to the bps rate.

Example: The following example selects com port 1, then sets the speed to 19200 bps.

```
Set Port 1
Set Speed 19200
```

See Also: Dial Host, Set Port

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set Timeout

Description: Use this command to set timeout values for subsequent Wait and Receive commands. This can be used to control script execution and prevent script "locking" in the event that an appropriate value is not received during the specified timeout period. If an On Timeout command has been specified, then it is executed if a timeout occurs during the execution of a Wait or Receive command.

Setting $Timeout to 0 disables the default timeout value for Wait and Receive commands. Set Timeout may be combined with an On Timeout command, for error-trapping and controlling script execution. See the example below.

Syntax: **Set Timeout** seconds

Parameters:

seconds	Integer literal or integer variable specifying the number of seconds before a timeout occurs.

Reserved Variables:

$Result	Set to 0.
$Error	Unchanged.
$Timeout	Set to the number of seconds.

Example: The following example ends the script when a timeout occurs.

```
Set Timeout 20          ;set timeout value
;specify where to branch if timeout occurs
On Timeout GoTo @timeout
...

@end:
Beep            ;let user know script has ended
End Terminal    ;exit to terminal mode

@timeout:
;a timeout occurred, end the script
Type String "^M^J(timeout)^M^J"
GoTo @end
```

See Also: On Timeout, Set Cancel, Set Disconnect, Wait and Receive commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Set Translation

Description: Use this command to activate/deactivate a previously loaded translation table. When on, all transmitted and received data is translated according to the currently loaded translation table. The name of the file containing the table is loaded along with the session configuration or with the Load Translation command. If you load a session configuration after this command, the active/inactive state is reset to that in the configuration.

Syntax: **Set Translation On** | **Off**

Parameters:

On	Activates the translation table.	
Off	Deactivates the translation table.	

Reserved Variables:

$Result	Set to 0.	
$Error	Unchanged.	

Example: The following example loads and activates a translation table. Then it sends a translated string.

```
Load Translation "AW"
Set Translation On      ;activate translation
Send String "translate this"
```

The next example deactivates a translation table and sends a string without translation.

```
Set Translation Off
Send String "do not translate"
```

See Also: Load Translation

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

StrCat

Description: Use this command to append one string to the end of another. You may want to trim leading and trailing blanks (using Trim) before using this command. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **StrCat** string1 string2

Parameters:

string1	String variable containing the string to which string2 will be added. After the command is executed, it contains the result of concatenating the two strings.
string2	String literal or string variable to be added to the end of string1.

Reserved Variables:

$Result	Set to the length of the resulting string. The quotation marks surrounding the string are not counted. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example concatenates "ABCDE" to string1 and sets $Result to 10.

```
String string1[15]
string1 = "12345"
;string1 becomes "12345ABCDE"
StrCat string1 "ABCDE"
```

See Also: Lower, Set IgnoreCase, StrCmp, StrLen, StrSet, SubStr, Trim, Upper

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

StrCmp

Description: Use this command to compare, character by character, the contents of two strings for equality. The $IgnoreCase reserved variable is used to determine if case sensitivity is enabled. The case of the characters is ignored if Set IgnoreCase has been set to on. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **StrCmp** string1 string2 [length]

Parameters:

string1	First string literal or string variable to be compared.
string2	Second string literal or string variable to be compared.
length	(Optional) integer literal or integer variable containing the number of characters to be compared. The default is the length of the shorter of the two strings.

Reserved Variables:

$Result	Set to 0 if the strings are identical. If string1 precedes string2 in ASCII order, $Result is set to -1; if string1 follows string2 in ASCII order, $Result is set to 1.
$Error	Standard errors. See Appendix B, “Error Messages.”

Example: The following example compares unequal strings. $Result becomes -1 because "ABC" precedes "DEF" in ASCII order.

```
StrCmp "ABC" "DEF"    ;$Result is set to -1
```

The next example compares unequal strings. $Result becomes 1 because "123" follows "12" in ASCII order.

```
String S1[3]
String S2[3]
S1 = "123"
S2 = "12"
StrCmp S1 S2          ;$Result is set to 1
```

The last example compares strings that are equal when the case of the characters is ignored. $Result becomes 0 because they are equal.

```
Set IgnoreCase On
StrCmp "ABC" "abc"    ;$Result is set to 0
```

See Also: Lower, Set IgnoreCase, StrCat, StrLen, StrSet, SubStr, Trim, Upper

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

StrLen

Description:	Use this command to determine the length of a string; for example, a password or ID specified by the user. The quotation marks surrounding the string are not counted. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.	
Syntax:	**StrLen** source [length]	
Parameters:	source	String literal or string variable whose length is to be determined.
	length	(Optional) integer variable that contains the number of characters in the string. Its value is also stored in $Result.
Reserved Variables:	$Result	Set to the length of the string. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example determines that "MICKEY" has six characters.

```
String s1[25]
Integer length
s1 = "MICKEY"
StrLen s1 length        ;length and $Result are set to 6
```

See Also: Lower, Set IgnoreCase, StrCat, StrCmp, StrSet, SubStr, Trim, Upper

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

StrSet

Description: Use this command to fill a string with the specified number of the specified character. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **StrSet** result_string length character

Parameters:

result_string — String variable to be created.

length — Integer literal or integer variable containing the number of characters to put in result_string. This becomes the length of the string.

character — Integer literal or integer variable indicating the ASCII value of the character, or string literal or string variable indicating the character itself. If a string literal or string variable is specified, only the first character in the string is used.

Reserved Variables:

$Result — Set to 0. If an error occurs, set to the value of $Error.

$Error — Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example creates a string (S1) containing three D's.

```
StrSet S1 3 "D"        ;s1 is set to DDD
```

The next example creates a string containing one A because 65 is the ASCII character for uppercase A.

```
StrSet S2 1 65
```

See Also: Lower, Set IgnoreCase, StrCat, StrCmp, StrLen, SubStr, Trim, Upper

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

SubStr

Description: Use this command to extract a portion of a string (called a substring) and store it. You specify the starting position within the original string and the number of characters to copy. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **SubStr** string start length substring

Parameters:

string	String literal or string variable from which a portion is to be copied.
start	Integer literal or integer variable specifying the starting position within the first string. The first character in a string is in position 1.
length	Integer literal or integer variable specifying how many characters to copy.
substring	String variable that will contain the copied substring.

Reserved Variables:

$Result	Set to the length of the resulting string. The quotation marks are not counted in the length.
$Error	Standard errors. See Appendix B, “Error Messages.”

Example: In the following example, string1 becomes "BCD" and $Result becomes 3.

```
String string1[50]
SubStr "ABCDEF" 2 3 string1
```

See Also: Lower, Set IgnoreCase, StrCat, StrCmp, StrLen, StrSet, Trim, Upper

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Terminal

Description:	Use this command to go into terminal emulation mode. The last terminal emulation, port, speed, parity, recording status, and so forth remain in effect. When terminal mode is ended, control returns to the current script at the next command.
Syntax:	**Terminal**
Reserved Variables:	$Result Set to 0.
$Error	Unchanged.
Example:	The following example goes into terminal emulation mode.
	Terminal
See Also:	End Terminal

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Trim

Description:	Use this command to remove, or trim, leading and trailing spaces and control codes from the specified string. It trims all ASCII characters with an ASCII value less than 33. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.	
Syntax:	**Trim** string	
Parameters:	string	String variable to be trimmed.
Reserved Variables:	$Result	Set to the length of the resulting string. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example trims the spaces from both ends of the S1 string.

```
String string1[10]
string1 = "      ABC      "
Trim string1    ;string1 is set to "ABC"
```

See Also: Lower, Set IgnoreCase, StrCat, StrCmp, StrLen, StrSet, SubStr, Upper

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Type File

Description:	Use this command to display the contents of a file on the screen. To make the scrolling of the output pause, the user can press Ctrl+S. To resume scrolling, the user can press any key. The output stops if the previous Set Cancel command was set to on and the user presses Esc. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.	
Syntax:	**Type File** filename	
Parameters:	filename	String literal or string variable containing path and name of file whose contents are to be displayed.
Reserved Variables:	$Result	Set to 0. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, “Error Messages.”
Example:	The following example displays the contents of the AUTOEXEC.BAT file on the screen. Type File "c:\autoexec.bat"	
See Also:	Type Line, Type String	

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Type Line

Description: Use this command to display a line on the screen. For example, you can display comments about the current script operation. Type Line appends a carriage return/linefeed combination ($TLineEnd) to the end of the string. The $TLineEnd variable can be changed to alter line spacing. For example, if $TLineEnd is set to "^M^J^J", lines will be ended with two carriage returns, creating double-spaced lines.

When the string you are displaying exceeds the number of columns available in the current row, the remainder of the string automatically wraps to the next line. If the cursor is currently on a row near the bottom of the screen, and the string exceeds the number of rows available, then the remainder of the string is wrapped beginning at row 1, column 1.

This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Type Line** string

Parameters:	string	String literal or string variable to be displayed.
Reserved Variables:	$Result	Set to the number of characters displayed. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example shows each of three comments on separate double-spaced lines.

```
$TLineEnd = "^M^J^J"
Type Line "COMMENT ONE"
Type Line "COMMENT TWO"
Type Line "COMMENT THREE"
```

See Also: Type File, Type String

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Type String

Description: Use this command to display a string on the screen. For example, you can display comments about the current script operation. Type String has the same effect as the Type Line command if the string is followed by the carriage return/linefeed combination (^M^J).

When the string you are displaying exceeds the number of columns available in the current row, the remainder of the string is automatically wrapped to the next line. If the cursor is currently on a row near the bottom of the screen, and the string exceeds the number of rows available, then the remainder of the string is wrapped beginning at row 1, column 1.

If you want to insert a string rather than type over video space or other characters, use Insert Char first. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Type** [**String**] string

Parameters:

String	(Optional) reserved word; the command may be shortened.	
string	String literal or string variable to be displayed.	

Reserved Variables:

$Result	Set to the number of characters displayed. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, “Error Messages.”

Example: The following example displays "one, two, three" followed by a carriage return/linefeed. Each Type (String) command starts where the previous one left off.

```
String string1
string1 = "one, "
Type string1
Type "two, "
Type "three^M^J"
```

The next example moves the cursor to an appropriate position and inserts a message in a previously typed line.

```
Cursor Left 10
Wait Time 2
;prepare for insertion
Insert Char 21
;insert the characters
Type String "Waiting for signal..."
```

See Also: Type File, Type Line, Insert Char

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Upper

Description:	Use this command to convert all lowercase characters in a string to uppercase. Characters that are already uppercase are not affected. This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.	
Syntax:	**Upper** string	
Parameters:	string	String literal or string variable to be converted.
Reserved Variables:	$Result	Set to 0.
	$Error	Unchanged.
Example:	The following example converts string1 from "Abc" to "ABC", then displays it.	
	String string1[3]	
	string1 = "Abc"	
	Upper string1	
	Type string1	
See Also:	Lower, Set IgnoreCase, StrCat, StrCmp, StrLen, StrSet, SubStr, Trim	

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Wait Carrier

Description: This command applies only to serial connections. Use it to wait for the data-carrier-detect signal (CD or DCD), signifying that a connection has been established. If an On Timeout command is active, it specifies timeout handling for this command.

Syntax: **Wait Carrier** [timeout]

Parameters:

timeout	(Optional) integer literal or integer variable specifying the maximum number of seconds to wait. The default is the number of seconds specified by the Set Timeout command.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example dials Office PC and waits two minutes for a connection. It will redial a total of nine times if necessary.

```
;show command processing on the remote
Set Echo On
;declare and initialize a counter
Integer count
count = 0

@retry:          ;the retry loop label
count = count + 1          ;increment counter
Dial Host "Office PC"      ;dial host
On Timeout GoTo @timeout          ;set error handling
;wait two minutes for carrier detection
Wait Carrier 120
;display status on remote PC
Type "Connection established..."
...

@timeout:
Type Line "No carrier..." ;display status
;loop if fewer than ten retries have occurred
If (count < 10) GoTo @retry
Type "Too many retries, try later..."
End;end session
```

See Also: Set Cancel, Set Disconnect, Set Timeout, other Wait commands

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Wait Receive

Description: Use this command in network connections to replace the Wait Carrier command. It waits for any character to be received from the communications device. If an On Timeout command is active, it specifies timeout handling for this command.

Syntax: **Wait Receive** [timeout]

Parameters:

timeout	(Optional) integer literal or integer variable specifying the maximum number of seconds to wait. The default is the number of seconds specified by the Set Timeout command.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example waits 60 seconds for a character to be received from the host.

```
;show command processing on the remote
Set Echo On
Set Timeout 60          ;establish a default timeout
;declare and initialize a counter
Integer count
count = 0

@retry:         ;the retry loop label
count = count + 1       ;increment counter
Dial Host "Office PC"   ;dial host
Wait Receive  ;wait 60 seconds for any character
;branch depending on the value for $Error
;these $Error values are for DOS script
If ($Error == -4) GoTo @timeout
If ($Error == -1) GoTo @canceled
If ($Error != 0) GoTo @another_error
...

@end:
End;end session

@timeout:
Type Line "A timeout has occurred..."
;retry or end session
If (count < 10) GoTo @retry
Type "Too many retries, try later..."GoTo @end
```

```
@canceled:
;end session
Type Line "Operator abort..."
GoTo @end

@another_error:
;end session
Type Line "Error occurred..."
GoTo @end
```

See Also: Set Cancel, Set Disconnect, Set Timeout, other Wait commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Wait Silence

Description:	Use this command to establish a received data timeout for unattended file transfers. It waits until it receives a specified number of seconds of communications silence. If an On Timeout command is active, it specifies timeout handling for this command.	
Syntax:	**Wait Silence** [timeout]	
Parameters:	timeout	(Optional) integer literal or integer variable representing length of silence to wait. The default is the number of seconds set by the Set Timeout command.
Reserved Variables:	$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."	

Example

The following example transfers the C:\FILE1 file from the host to the remote, waits for a minute of inactivity, then transfers C:\FILE2 to the host.

```
$Error = 0      ;reset $Error
On Error GoTo @error   ;handle errors
Send String "AWSEND C:\FILE1 to remote"
Receive File "C:\FILE1"
Wait Silence 60        ;wait for 1 minute of silence
Send String "AWSEND C:\FILE2 to host"
Send File "C:\FILE2"
```

See Also: Set Cancel, Set Disconnect, Set Timeout, other Wait commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Wait String

Description: Use this command to wait for a specified string, for example, a password prompt, to be received. Because scripts begin execution as soon as the remote calls a host, password prompts on the remote PC can be handled by script commands. If an On Timeout command is active, it specifies timeout handling for this command.

By default, strings in pcAnywhere are not case-sensitive. You may want to use the Set IgnoreCase command to enable case-sensitivity before prompting the user for a password.

NOTE: Wait String is usually preceded by a Wait Carrier or Wait Receive command to ensure appropriate script execution.

Syntax: **Wait String** string [timeout]

Parameters:

string	String literal or string variable to wait for.
timeout	(Optional) integer literal or integer variable specifying the number of seconds to wait. The default is the number of seconds set by the Set Timeout command.

Reserved Variables:

$Result	Set to 0. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example waits ten seconds for a password, makes sure the case of the password's characters are ignored, and contains error handling in case no password is given.

```
$Error = 0      ;reset $Error to zero
On Error GoTo @nopassword       ;set-up error handling
Set IgnoreCase On       ;ignore case
Set Echo On
;wait up to 10 seconds for password
Wait String "Enter password:" 10
...

@nopassword:
;if no password received, hand up the phone and
;try again
Type Line "No password prompt."
Hang
GoTo @retry
```

See Also: Set Cancel, Set Disconnect, Set Echo, Set Timeout, other Wait commands

{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Wait Time

Description:	Use this command to wait for the specified number of seconds. For example, use the waiting period to display received data (if echo is on), record received data (if recording is on), or check to see if the data matches the specified On Receive string (if enabled).	
Syntax:	**Wait** [**Time**] seconds	
Parameters:	**Time**	(Optional) reserved word; the command may be shortened.
	seconds	Integer literal or integer variable specifying the number of seconds to wait.
Reserved Variables:	$Result	Set to 0. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, "Error Messages."
Example:	The following example prepares for a session, then waits 60 seconds before proceeding with the script.	
	Set Echo On ;display received data	
	Set Record On ;record received data	
	Wait Time 60 ;wait for 60 seconds	
	...	
See Also:	Set Cancel, Set Disconnect, Set Timeout, other Wait commands	

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Wait Until

Description: Use this command to wait until a specific time and date occurs. For example, you can delay script completion until after normal working hours or until a time when controlling a non-dedicated host will not interfere with other operations. Place the command at the beginning of the script, before a Dial Host command, and before any loop that redials (see the example for the Wait Carrier command).

Current time is read from the system clock. (The DOS command Time displays the current system time.) You can change the system date and time from the Windows Control Panel using the Date/Time option or from the DOS prompt using the DOS date and time commands.

Syntax: **Wait Until** time [date]

Parameters:		
	time	Integer literal or integer variable specifying time to wait for, in the 24-hour format HHMM.
	date	(Optional) integer literal or integer variable specifying date to wait for, in the format YYMMDD. The default is the current date.

Reserved Variables:		
	$Result	Set to 0. If an error occurs, set to the value of $Error.
	$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example waits until 10:40 P.M. BEFORE COMPLETING THE SCRIPT.

```
Wait Until 2240
```

See Also: Set Cancel, Set Disconnect, Set Timeout, other Wait commands

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Write Line

Description: This command applies to the remote PC only. Use it to write a string to a file and append a carriage return/linefeed combination ($TLineEnd). For example, you can create comments in log files which permanently record unattended script operations. The values of string variables may also be stored in a file for later review.

The $TLineEnd variable may changed to alter line-spacing. For example, if $TLineEnd is set to "^M^J^J", lines will be ended with two carriage returns, creating double-spaced lines.

This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Write Line** file_number string

Parameters:

file_number	Integer literal used by Open or Create command to identify the file. Valid identifiers are 1 through 9.
string	String literal or string variable to be written to the file.

Reserved Variables:

$Result	Set to the number of bytes written. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, ‘‘Error Messages.’’

Example: The following example opens a log file, positions its file pointer, writes three double-spaced lines, then closes the log file.

```
$TLineEnd = "^M^J^J"   ;start double-spacing
Open 1 "logfile" "WC"
Seek 1 0 2
Write Line 1 "First comment"
Write Line 1 "Second comment"
Write Line 1 "Third comment"
Close 1
```

See Also: Read Line, Read String, Write String

Script Editor Commands
{button A,JI(`',`a')} {button B,JI(`',`b')} {button C,JI(`',`c')} {button D,JI(`',`d')} {button E,JI(`',`e')} {button F,JI(`',`f')} {button G,JI(`',`g')} {button H,JI(`',`h')} {button I,JI(`',`i')} {button K,JI(`',`k')} {button L,JI(`',`l')} {button M,JI(`',`m')} {button O,JI(`',`o')} {button P,JI(`',`p')} {button R,JI(`',`r')} {button S,JI(`',`s')} {button T,JI(`',`t')} {button U,JI(`',`u')} {button W,JI(`',`w')}

Write String

Description: This command applies to the remote PC only. Use it to write a string to a file without appending a carriage return/linefeed. For example, you can create comments in log files which permanently record unattended script operations. The values of string variables may also be stored in a file for later review.

This command will have the same effect as the Write Line command if the string is followed by a carriage return/linefeed combination, ^M^J.

This command can be used in scripts that perform remote control sessions so long as the command is not executed during the remote control session.

Syntax: **Write String** file_number string [length]

Parameters:

file_number	Integer literal used by Open or Create command to identify the file. Valid identifiers are 1 through 9.
string	String literal or string variable to be written to the file.
length	(Optional) integer literal or integer variable limiting the number of characters to write. The default is the length of the string.

Reserved Variables:

$Result	Set to the number of bytes written. If an error occurs, set to the value of $Error.
$Error	Standard errors. See Appendix B, "Error Messages."

Example: The following example opens a file, positions its file pointer, writes two strings side-by-side, then closes the file.

```
Open 1 "myfile.dat" "WC"
Seek 1 0 2
Write String 1 "First this; "
Write String 1 "then that"
Close 1
```

See Also: Read Line, Read String, Write Line

Script Editor Commands Listed by Task **{button Click to switch to alphabetical list.,JI(`',`script_editor_commands_alphabet')}**

Click any task to see related commands you can include in a script:

- Automating Remote Control Sessions
- Loading or Modifying Configuration Information
- Controlling Script Processing
- Interacting with User
- Transmitting and Receiving Data
- Managing Files and Directories
- Reading/Writing to Files
- Performing String Operations
- Printing
- Managing the Display
- Miscellaneous Commands

Automating Remote Control Sessions **{button Click to switch to alphabetical list.,JI(`',`script_editor_commands_alphabet')}**

Click any command to view syntax, description, and an example:

- Session Delay
- Session Dial
- Session End
- Session ExitMode
- Session OnError
- Session Overwrites
- Session Retry
- Session Timeout
- SessOpr Host Run
- SessOpr Host Send
- SessOpr Remote Run
- SessOpr Remote Send

Loading or Modifying Configuration Information **{button Click to switch to alphabetical list.,JI(`',`script_editor_commands_alphabet')}**

Click any command to view syntax, description, and an example:

- Answer
- Dial Host
- Dial Number
- Dial OnlSvc
- Emulate
- Load FKeys
- Load HostInfo
- Load OnlSvcInfo
- Load Translation
- Set DTR
- Set Echo
- Set Flow
- Set Parity
- Set Port
- Set Protocol
- Set Record
- Set RTS
- Set Speed
- Set Translation

Controlling Script Processing **{button Click to switch to alphabetical list.,JI(`',`script_editor_commands_alphabet')}**

Click any command to view syntax, description, and an example:

- End
- End Menu
- End Terminal
- Exit
- GoSub
- GoTo
- Hang
- If...Then or If...GoTo
- Link
- On Cancel
- On Disconnect
- On Error
- Return
- Script
- Set Cancel
- Set Disconnect
- Terminal

Interacting with User **{button Click to switch to alphabetical list.,JI(`',`script_editor_commands_alphabet')}**

Click any command to view syntax, description, and an example:

- Beep
- Input
- Input Key
- Keyboard Flush
- Keyboard Hit
- Message Box
- Type File
- Type Line
- Type String

Transmitting and Receiving Data **{button Click to switch to alphabetical list.,JI(`',`script_editor_commands_alphabet')}**

Click any command to view syntax, description, and an example:

- Break
- On Receive
- On Timeout
- Receive Char
- Receive Clear
- Receive File
- Receive Line
- Receive String
- Reset
- Send Char
- Send Clear
- Send File
- Send Line
- Send String
- Set CharDelay
- Set Timeout
- Wait Carrier
- Wait Receive
- Wait Silence
- Wait String
- Wait Time
- Wait Until

Managing Files and Directories **{button Click to switch to alphabetical list.,JI(`',`script_editor_commands_alphabet')}**

Click any command to view syntax, description, and an example:

- CD
- Copy
- Del
- Dir
- Find First
- Find Next
- Get Environment
- Get File Attr
- Get File Date
- Get File Size
- Get File Time
- Get Free Disk
- MD
- RD
- Ren
- Set File Attr
- Set File Date
- Set File Size
- Set File Time

Reading/Writing to Files **{button Click to switch to alphabetical list.,JI(`',`script_editor_commands_alphabet')}**

Click any command to view syntax, description, and an example:

- Open
- Close
- Read Line
- Read String
- Seek
- Write Line
- Write String

Performing String Operations **{button Click to switch to alphabetical list.,JI(`',`script_editor_commands_alphabet')}**

Click any command to view syntax, description, and an example:

- Decrypt
- Encrypt
- Index
- Lower
- Set IgnoreCase
- StrCat
- StrCmp
- StrLen
- StrSet
- SubStr
- Trim
- Upper

Printing **{button Click to switch to alphabetical list.,JI(``,`script_editor_commands_alphabet')}**

Click any command to view syntax, description, and an example:

- Print File
- Print Line
- Print String
- Printer

Managing the Display **{button Click to switch to alphabetical list.,JI(`',`script_editor_commands_alphabet')}**

Click any command to view syntax, description, and an example:

- Clear BOL
- Clear BOP
- Clear EOL
- Clear EOP
- Clear Screen
- Cursor Block
- Cursor Down
- Cursor Home
- Cursor Left
- Cursor Line
- Cursor Off
- Cursor Position
- Cursor Restore
- Cursor Right
- Cursor Save
- Cursor Up
- Delete Char
- Delete Line
- Insert Char
- Insert Line
- Screen Restore
- Screen Save
- Set Attribute
- Set Palette
- Set Quiet

Miscellaneous Commands **{button Click to switch to alphabetical list.,JI(`',`script_editor_commands_alphabet')}**

Click any command to view syntax, description, and an example:

- Description
- Let
- Run

-A-

ASCII

assignment operator

-B-

baud rate

binary operator

bitwise And

bitwise Or

bps

break length

bug

-C-

case sensitivity

clipboard

command

comment

communications protocol

compile

compile time

constant

control-menu box

control menu

-D-

data rate

directive

-E-

equality operator

exclusive Or

-F-

file transfer

flow control

-G-

ASCII

Standardized codes used to represent characters, commands and other data, thereby enabling different programs, computer systems and hardware devices to exchange information; stands for American Standard Code for Information Interchange.

assignment operator

An operator that assigns the value on the right side of the equal sign to the variable on the left side of the equal sign. The pcAnywhere assignment operator is the equal sign (=). The assignment operator can be used with integers or strings. Unless otherwise noted, you can also assign values to the script language's reserved variables.

binary operator

An operator having two terms. pcAnywhere can use binary arithmetic operators, relational operators and logical operators. Compare to unary operator.

bitwise And

Usually used in masking, a bit by bit Anding of two binary values.

bitwise Or

Usually used in masking, a bit by bit Oring of two binary values.

bug
Any problem that exists within a program.

case sensitivity
Differentiation between uppercase letters and lowercase letters. Case sensitivity means that a string such as "Password" does not match "PASSWORD." On the other hand, case insensitivity means that "Dial Number" matches "DIAL NUMBER," "dial number" or any other combination of upper and lowercase letters.

clipboard
A temporary storage area for cut or copied text or graphics. The clipboard can be used as a holding pen for information you want to pass between Windows programs.

You can place data in the clipboard by using the Cut or Copy command of the Edit menu. Insert data from the clipboard into a document by using the Edit Paste command.

command

Commands are instructions to pcAnywhere. Each command has a name consisting of one or more reserved words, followed by zero or more parameters. Parameters provide additional information about how the instruction is to be executed. Depending on the instruction, a parameter may be an integer or string variable or constant, or an arithmetic expression.

comment

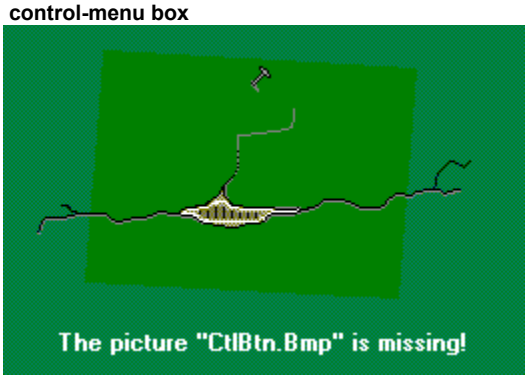
A string which is ignored by the pcAnywhere compiler and is not executed. Use comments in your script files to provide documentation and to serve as reminders during subsequent modifications to the script. A comment is designated by a semicolon (;) that is not within delimiting quote marks. pcAnywhere considers all text to the right of the semicolon as a comment. Comments can be entered on a line by themselves, or added to the end of any other command line.

compile
The process of converting a high-level script into a low-level set of commands which can be executed, or run. Syntax errors are discovered when a script is being compiled.

compile time
Refers to the events which occur during the compilation process.

constant
A symbol for a fixed numeric or character value. pcAnywhere supports integer and string constants.

control-menu box



Click this button to open a menu which provides general control for the active window.

Double-click it to close the window.

Click it once to drop-down the application control-menu which lets you close, resize or move the window.

Keyboard users can press Alt+spacebar to access the control-menu.

For programs which support multiple documents (such as the pcAnywhere Editor), press Alt+hyphen to drop-down the document control-menu for the active editing window.

control menu

Click  the control-menu box (Alt+spacebar) to drop-down the control menu. This menu provides a way for keyboard users to move and resize the window.

directive

A directive is a script file statement which helps pcAnywhere interpret your intentions. Directives in the script language are the integer and string declarations.

equality operator
The symbol testing whether the values on each side of it are equal. pcAnywhere's equality operator is a double-equal sign (==). **See also** relational operator.

exclusive Or

A logical comparison in which only *one* of the two conditions being compared is true.

global variable

Global variables are variables whose values are remembered and remain accessible while you are running linked or nested scripts. These variables retain their values for the duration of the script execution. pcAnywhere contains 127 global variables, named $1 through $127.

identifier

Identifiers are the names supplied for variables, parameters, labels, and commands in scripts. An identifier is a single letter, or else a sequence of letters and/or digits that begins with a letter.

Only the first eight characters of an identifier determine its uniqueness within the script.

integer
Any whole (non-fractional) number, such as 0, 1 or 9275.

integer constant

Any value built from the digits 0 through 9. Unlike string constants, integer constants are not delimited by quote marks. Integer constants in pcAnywhere can have a value from -2,147,483,648 to 2,147,483,647 and cannot contain any commas or decimal points.

key word

Key words are identifiers that have special meaning to pcAnywhere and cannot be used as user-defined variable names. pcAnywhere key words include the titles of all commands, reserved variables and global variables.

label

An identifier that serves as a destination address or marker. In the script language, a label is used with the Goto and Gosub commands. The destination to which the Goto or Gosub command is to transfer control is indicated by a colon following the label name.

loop

A portion of a program that gets repeated. The script language uses Goto statements, in conjunction with If...Then statements, to set up loops.

minimize and **maximize**

When you click  the Minimize button (Alt+Spacebar,N) located near the top-right corner of a window, that window shrinks into an icon and appears on your taskbar.

You can click the icon to restore it to its previous size.

When you click  the Maximize button (Alt+Spacebar,X) located at the top-right corner of a window, that window instantly expands to fill the screen.

You can click  the Restore button to restore the window to its previous size.

- You can also maximize or restore a window by double-clicking its title bar.

nest

A nested expression is an expression whose result is used as an operand in another expression. The expressions in the innermost parentheses are evaluated first.

For example, the command

```
IF ((a == b) && (c != d)) THEN GOTO @continue
```

contains two nested expressions: (a==b) and (c!= d). These expressions are evaluated first, and their results are used as the operands for the **logical And (&&)** expression.

operator

A symbol within a statement that denotes a type of operation. For example, the equal sign (=) is the assignment operator, and the double-equal sign (==) is the equality relational operator.

parameter

Information that is supplied to a command. Parameters follow the command name and are separated by spaces. Some parameters must be specified; others are optional.

relational operator

A binary operator for comparing the values on either side of the symbol. For example, in the script language > is the relational operator for "greater than" and == is the equality operator (the relational operator for "equal to"). The script language can use relational operators with either integers or strings.

reserved variable

Reserved variables are variables that have special meaning to pcAnywhere. They may contain, for example, current parameter settings, results of operations, or the current date and time. You can use reserved variables anywhere that variables of the appropriate type (integer or string) are allowed.

return value

The information returned by a command. Many commands store the return value in the $RESULT reserved variable; others, such as Find First, allow you to specify a user variable to store it.

run time
Refers to the events which occur while the program is actually executing.

script

A standard ASCII test file containing statements of the script language for pcAnywhere. A script usually consists of instructions expressed using the script language's rules and syntax, combined with simple control structures.

pcAnywhere source scripts have an extension of .SCR.

Compiled, executable pcAnywhere scripts have extensions of .SCX.

You can create a pcAnywhere script using the pcAnywhere Editor or any other standard ASCII text editor.

scope

The set of rules governing when and how a variable may be accessed. These rules determine when the variable name is valid; that is, when it can be used in a script. Since script files can call other script files to be run, the variables named in the first script may not be accessible to the called script.

statement

A program unit consisting of a command, an assignment operator or a combination of both. A statement is equivalent to a script file line.

string
Any series of characters surrounded by a pair of single or double quotation mark delimiters.

string constant
A value that is delimited by quotation marks.

string variable

A placeholder for text. For example, in the statement prompt = "Please click Yes, No or Cancel" the variable named prompt is a string variable.

syntax
The set of rules governing the way a command can be used.

syntax error

Syntax errors are detected during compilation and occur when you make a mistake entering a command, such as not enclosing a string in quotes, or specifying the wrong number of parameters. The syntax errors are written to a file with the same source filename and the extension .ERR. You can use the pcAnywhere Editor to view the .ERR file, make corrections to the script and attempt compilation again.

unary operator
An arithmetic operator having only one term. The script language's unary operator is the minus sign (-). Compare to binary operator.

user variable

User variables are those variables that you define for your own use during the running of a single script. At the end of script execution, the values stored in these variables are forgotten.

variable

Variables identify data which may change during the running of a script or each time a script is run. **See also** global variable, reserved variable and user variable.

wildcard

A symbol that enables multiple matching values to be returned based on a shared feature. The script language has two wildcards: the question mark (?) and the asterisk (*). The question mark stands for any single character, and the asterisk stands for any character string of any length. For example, the file specification *.* would return all files, regardless of their filenames; the file specification *.SC? would return all filenames having a three-character extension beginning with SC (COMPUSRV.SCR, COMPUSRV.SCX, etc.).

Text Editor Functions

about

Displays the editor's version number, copyright notice and miscellaneous system information.

Keyboard shortcut: Alt+V.

Text Editor Functions

backspace

If text is selected, deletes it; otherwise, deletes the character to the left of the cursor.

If the cursor is at the beginning of the line, appends the current line to the previous line.

Keyboard shortcut: Backspace

Text Editor Functions

batch_help

Displays help text on each script language command and lets you select commands to enter into your current file.

Keyboard shortcut: Alt+R

Text Editor Functions

beginning_of_buffer

Moves the cursor to the beginning of the document in the active window.

Keyboard shortcut: Ctrl+Home

Text Editor Functions

beginning_of_line

Moves the cursor to the beginning of the current line.

Keyboard shortcut: Home

Text Editor Functions

bottom_of_window

Moves the cursor to the bottom line of the window.

Keyboard shortcut: Ctrl+PageDown

Text Editor Functions

cascade

Resizes and rearranges all open windows in an overlapping pattern. Same as the Cascade command on the Window menu.

Keyboard shortcut: Alt+W, then C

Text Editor Functions

change_case

Within the selected block, changes the case of all letters, making uppercase letters into lowercase, and lowercase letters into uppercase.

- To assign a keyboard shortcut to this Text Editor function, use the Key Assignments option on the Options menu.

Text Editor Functions

close_window

Closes the active window. If you have made changes to the file, the pcAnywhere Editor prompts you to save them if you have not already done so.

Same as the Close command on the File menu. You can also close a window by double-clicking its document control-menu button.

Keyboard shortcut: Ctrl + F4

Text Editor Functions

compare

Prompts for two filenames, then compares them line-by-line.

Same as the Compare command on the File menu.

Keyboard shortcut: Alt+F, then E

Text Editor Functions

copy

Copies the selected text to the clipboard.

If no text is selected, but the Editor Preferences dialog option Cut/Copy line if no text is selected is checked, then this copies the entire line to the clipboard.

Same as the Copy command on the Edit menu.

Keyboard shortcut: Ctrl + Insert

Text Editor Functions

copy_block

Copies the selected text to the clipboard.

This variation of the copy command has no effect if no text is selected.

- To assign a keyboard shortcut to this Text Editor function, use the Key Assignments option on the Options menu.

Text Editor Functions

copy_line

Copies the current line to the clipboard.

Text Editor Functions

cursor_down

Moves the cursor down one line.

Keyboard shortcut: DownArrow

Text Editor Functions

cursor_left

Moves the cursor one character to the left or, if in the leftmost column, to the end of the line above.

Keyboard shortcut: LeftArrow

Text Editor Functions

cursor_right

Moves the cursor one character to the right or, if at the end of the line, to the beginning of the line below.

Keyboard shortcut: RightArrow

Text Editor Functions

cursor_up

Moves the cursor up one line.

Keyboard shortcut: UpArrow

Text Editor Functions

cut

Removes the selected text, placing it in the clipboard.

If no text is selected, but the editor's Preferences dialog option Cut/Copy Line if No Text is Selected is checked, then this cuts the entire line from the document and copies it to the clipboard.

Same as the Cut command on the Edit menu.

Keyboard shortcut: Shift + Del

Text Editor Functions

cut_block

Removes the selected text, placing it in the clipboard.

This variation of the cut command has no effect if no text is selected.

Text Editor Functions

cut_line

Removes the current line, placing it in the clipboard.

Text Editor Functions

delete

Deletes the selected text. If no text is selected, deletes the character at the cursor.

The text can be recovered using the undo function.

Keyboard shortcut: Del

Text Editor Functions

delete_line

Deletes the current line.

The line is **not** copied to the clipboard, but can be recovered using the undo function.

Keyboard shortcut: Alt + D

Text Editor Functions

delete_to_eol

Deletes all text from the cursor to the end of the current line.

The text can be recovered using the undo function.

Keyboard shortcut: Alt + K

Text Editor Functions

delete_word_left

If the cursor is in a word, deletes the text from the cursor to the beginning of the word.

If the character to the left of the cursor is a space or tab, deletes the text from the cursor to the previous non-blank character.

If the character to the left of the cursor is a delimiter other than space or tab, deletes that delimiter.

Keyboard shortcut: Ctrl + BackSpace

Text Editor Functions

delete_word_right

If the cursor is in a word, deletes the text from the cursor to the end of the word.

If the character to the right of the cursor is a space or tab, deletes the text from the cursor to the next non-blank character.

If the character to the right of the cursor is a delimiter other than space or tab, deletes that delimiter.

Keyboard shortcut: Ctrl + Del

Text Editor Functions

document_preferences

Displays the Document Preferences dialog box, which lets you select options for the file in the active window.

Same as the Document Preferences command on the Options menu.

Keyboard shortcut: F4

Text Editor Functions

editor_help

Accesses the editor's help system.

Keyboard shortcut: F1

Text Editor Functions

end_of_buffer

Moves the cursor to the end of the document in the active window.

Keyboard shortcut: Ctrl + End

Text Editor Functions

end_of_line

Moves the cursor to the end of the current line.

Keyboard shortcut: End

Text Editor Functions

enter

In insert mode, inserts a carriage return and line feed, then moves the cursor to the beginning of the next line (or positions the cursor below the first non-blank character in the previous line if Auto Indent is checked in the Document Preferences dialog box).

In overwrite mode, simply moves the cursor to the beginning of the next line.

Keyboard shortcut: Enter

Text Editor Functions

exit

Prompts you to save any modified files, then closes all windows and exits from the editor.

Same as the **Exit** command on the **File** menu.

Keyboard shortcut: Alt + F4

Text Editor Functions

exit_windows

Prompts you to save any modified files, then issues an ExitWindows call to end the current Windows session. The session is not terminated unless all applications agree to terminate.

- To assign a keyboard shortcut to this Text Editor function, use the Key Assignments option on the Options menu.

Text Editor Functions

find

Displays the Find dialog box, which lets you set search criteria for the active document and begins searching.

Same as the Find command on the Search menu.

Keyboard shortcut: Ctrl + S

Text Editor Functions

find_again

Continues a search you initiated using the find function.

Same as to the **Find Again** command on the **Search** menu.

Keyboard shortcut: Ctrl + A

Text Editor Functions

find_files_containing

Displays the Find Files Containing dialog box. Use this to set search criteria and begin searching for files. Use the list_files_containing function to see and open the found files.

Same as the Find Files Containing command on the Search menu.

Keyboard shortcut: Ctrl + F

Text Editor Functions

goto_line

Prompts for a line number, then moves the cursor to the specified line in the active document.

If any text is selected, the selection is extended to include the requested line.

Same as the Goto Line command on the Search menu.

Keyboard shortcut: Ctrl + G

Text Editor Functions

list_files_containing

Use after using the find_files_containing function to display a list of found files. You can then double click a file to open it and remove it from the list.

Same as the List Found Files command on the Search menu.

Keyboard shortcut: Ctrl + L

Text Editor Functions

lowercase

Converts all uppercase characters in the marked block to lowercase.

If no block is marked, acts on the character at the cursor.

- To assign a keyboard shortcut to this Text Editor function, use the Key Assignments option on the Options menu.

Text Editor Functions

new_file

Opens a blank untitled document window. Use Save or Save As... on the File menu to give it a name and save the contents to disk.

Same as the New command on the File menu.

Keyboard shortcut: Alt + F, then N

Text Editor Functions

new_window

Opens an additional window on the current active document.

Changes made to either window affect the same document. If you want two different versions, first, use the Save As... command of the File menu to make a duplicate, then open both files.

Same as the New Window command on the Window menu.

Keyboard shortcut: Alt + W, then N

Text Editor Functions

next_window

Activates the next window in the editor's circular list.

Same as the Next Window command on the document control-menu.

Keyboard shortcut: Ctrl + F6

Text Editor Functions

open_file

Prompts for a directory and filename and opens a window on that file.

Same as the Open command on the File menu.

Keyboard shortcut: F3

Text Editor Functions

page_down

Moves the cursor down one page; that is, the number of lines visible in the window.

Keyboard shortcut: PageDown

Text Editor Functions

page_up

Moves the cursor up one page; that is, the number of lines visible in the window.

Keyboard shortcut: PageUp

Text Editor Functions

paste

Inserts the contents of the clipboard at the cursor location.

Same as the Paste command on the Edit menu.

Keyboard shortcut: Shift + Insert

Text Editor Functions

play_macro

Replays the keystrokes and functions recorded by the most recent use of the record_macro function.

If no macro has been recorded, that is indicated on the status line.

Same as the Play Back Macro command on the Edit menu.

Keyboard shortcut: F8

Text Editor Functions

prev_window

Activates the previous window on the editor's circular list.

- To assign a keyboard shortcut to this Text Editor function, use the Key Assignments option on the Options menu.

Text Editor Functions

print

Prints the active document, using settings set from the Page Setup and Printer Setup commands under the File menu.

Same as the Print command on the File menu.

Keyboard shortcut: Alt + P

Text Editor Functions

record_macro

Starts recording keystrokes and editor functions. Recording continues until the next use of this function.

Same as the Record Macro/Stop Recording Macro commands on the Edit menu.

Keyboard shortcut: F7

Text Editor Functions

replace

Prompts for search and replace criteria and then replaces specified text in the active document, starting at the cursor location.

Same as the Replace command on the Search menu.

Keyboard shortcut: Ctrl + R

Text Editor Functions

restore_window

Resizes the active window to its user-configurable "intermediate" size.

Same as the Restore command on the document control-menu.

Keyboard shortcut: Ctrl + F5

Text Editor Functions

revert

Prompts for verification, then undoes all changes to the contents of the active document since the file was last saved.

Same as the Revert command on the File menu.

Keyboard shortcut: Alt + F, then V

Text Editor Functions

save_all

Saves the contents of all modified document to disk. For untitled windows, prompts for filenames.

Same as the Save All command on the File menu.

Keyboard shortcut: Alt + F, then L

Text Editor Functions

save_all_exit

Saves the contents of all modified document to disk, then ends the editor session.

Keyboard shortcut: Alt + X

Text Editor Functions

save_all_exit_windows

Saves the current contents of all modified documents to disk, then ends the current Windows session. The Windows session is not terminated unless all applications agree to terminate.

Text Editor Functions

save_file

Saves the document in the current active window to disk. If the window is untitled, you are prompted for a filename.

Same as the Save command on the File menu.

Keyboard shortcut: F2

Text Editor Functions

save_file_close_window

Saves the contents of the current active document to disk, then closes the active window. If the window is untitled, the pcAnywhere Editor prompts you for a filename.

- To assign a keyboard shortcut to this Text Editor function, use the Key Assignments option on the Options menu.

Text Editor Functions

select_all

Selects (highlights) the entire contents of the active document.

Same as the Select All command on the Edit menu.

Keyboard shortcut: Alt + E, then A

Text Editor Functions

select_char_left

Selects the character to the left of the cursor, or cancels the selection if that character is already selected.

If the cursor is at the beginning of a line, it moves to the end of the previous line.

Keyboard shortcut: Shift + LeftArrow

Text Editor Functions

select_char_right

Selects the character to the right of the cursor, or cancels the selection if that character is already selected.

If the cursor is at the end of a line, it moves to the beginning of the next line.

Keyboard shortcut: Shift + RightArrow

Text Editor Functions

select_line

Selects the current line.

Text Editor Functions

select_line_down

Extends the current selection down one line, or cancels the selection if that line is already selected.

Keyboard shortcut: Shift + DownArrow

Text Editor Functions

select_line_up

Extends the current selection up one line, or cancels the selection if that line is already selected.

Keyboard shortcut: Shift + UpArrow

Text Editor Functions

select_page_down

Extends the current selection down one page and scrolls the window.

Keyboard shortcut: Shift + PageDown

Text Editor Functions

select_page_up

Extends the current selection up one page and scrolls the window.

Keyboard shortcut: Shift + PageUp

Text Editor Functions

select_to_bol

Selects the text from the cursor to the beginning of the current line, or cancels the selection if the text is already selected.

Keyboard shortcut: Shift + Home

Text Editor Functions

select_to_end

Selects the text from the cursor to the end of the document in the active window.

Keyboard shortcut: Ctrl + Shift + End

Text Editor Functions

select_to_eol

Selects the text from the cursor to the end of the current line, or cancels the selection if the text is already selected.

Keyboard shortcut: Shift + End

Text Editor Functions

select_to_top

Selects the text from the cursor to the beginning of the document in the active window.

Keyboard shortcut: Ctrl + Shift + Home

Text Editor Functions

select_word

Selects the word containing the cursor. (You can also select a word by positioning the cursor on the word and double-clicking the primary mouse button.)

Text Editor Functions

select_word_left

Extends the current selection to the beginning of the word to the left of the cursor, or cancels the selection if the text is already selected.

Keyboard shortcut: Ctrl + Shift + LeftArrow

Text Editor Functions

select_word_right

Extends the current selection to the beginning of the word to the right of the cursor, or cancels the selection if the text is already selected.

Keyboard shortcut: Ctrl + Shift + RightArrow

Text Editor Functions

split_line

Inserts a line break at the cursor without advancing the cursor.

Keyboard shortcut: Ctrl + N

Text Editor Functions

stamp

Inserts the current date and time at the cursor location.

Same as the Time/Date command on the Edit menu.

Text Editor Functions

tab_right

In insert mode, inserts a tab character at the cursor (or inserts spaces, if Expand Tabs with Spaces in the Document Preferences dialog box is checked).

In overwrite mode, moves the cursor to the next tab position, as set in that same dialog.

Keyboard shortcut: Tab

Text Editor Functions

test_batch

Executes the script commands in the active editing window.

Text Editor Functions

tile

Resizes and rearranges all open windows to fit within the editor main window with no overlap.

Same as the Tile command on the Window menu.

Keyboard shortcut: Alt + W, then T

Text Editor Functions

to_bottom

Scrolls the active window to place the current line at the bottom of the window.

Keyboard shortcut: Ctrl + B

Text Editor Functions

to_center

Scrolls the active window to place the current line to the center of the window.

Keyboard shortcut: Ctrl + C

Text Editor Functions

to_top

Scrolls the active window to place the current line to the top of the window.

Keyboard shortcut: Ctrl + T

Text Editor Functions

toggle_backup

Toggles the Make Backup Files option in the Editor Preferences dialog box.

When the backup option is on and you save a file, the original file is retained, but with an extension of .BAK.

- To assign a keyboard shortcut to this Text Editor function, use the Key Assignments option on the Options menu.

Text Editor Functions

toggle_insert

Toggles between insert mode and overwrite mode. The current mode is displayed on the status line.

Keyboard shortcut: Insert

Text Editor Functions

toggle_wordwrap

Toggles into and out of word wrap mode. The current mode is displayed on the status line.

When word wrap is on, typing past the right margin causes the current word to be moved to the next lower line.

Same as the Word Wrap option on the Edit menu.

Keyboard shortcut: Ctrl + W

Text Editor Functions

top_of_window

Moves the cursor to the top visible line of the window.

Keyboard shortcut: Ctrl + PageUp

Text Editor Functions

undo

Reverses the effects of the a previous editing operation. Repeated use undoes up to 300 edit operations.

Same as the Undo command on the Edit menu.

Keyboard shortcut: Alt + BackSpace

Text Editor Functions

unmark_block

Deselects the selected text and returns the cursor to its location before the text was selected.

Keyboard shortcut: Esc

Text Editor Functions

uppercase

Converts all lowercase characters in the marked block to uppercase. If no block is marked, converts the character at the cursor.

- To assign a keyboard shortcut to this Text Editor function, use the Key Assignments option on the Options menu.

Text Editor Functions

window_down

Moves the cursor up one line and at the same time moves the text in the window down one line.

Keyboard shortcut: Ctrl + UpArrow

Text Editor Functions

window_up

Moves the cursor down one line and at the same time moves the text in the window up one line.

Keyboard shortcut: Ctrl + DownArrow

Text Editor Functions

word_left

Moves the cursor to the beginning of the previous word or to the end of the previous line, whichever comes first.

Keyboard shortcut: Ctrl + LeftArrow

Text Editor Functions

word_right

Moves the cursor to the beginning of the next word or to the end of the line, whichever comes first.

Keyboard shortcut: Ctrl + RightArrow

Text Editor Functions

wrap_para

Reformats the current paragraph within the margins of the active document. Margins are set using the Document Preferences command on the Options menu.

Same as the Wrap Paragraph command on the Edit menu.

Keyboard shortcut: F12

Text Editor Functions

zoom_window

Resizes the active document window to its maximum possible size within the editor's window.

Same as the Maximize command on the document control-menu. You can also maximize and restore a window by double-clicking its title bar.

Keyboard shortcut: Ctrl + F1

Text Editor Functions: Functional Groups

Use Text Editor functions as you are creating scripts. Text Editor functions are **not** Editor commands and should not be included in scripts.

Click one of the following groups to see a listing of Text Editor functions:

Cursor Motion Text Editor Functions

Click an item below for more information.

beginning_of_buffer
beginning_of_line
bottom_of_window
cursor_down
cursor_left
cursor_right
cursor_up
end_of_buffer
end_of_line
enter
goto_line
page_down
page_up
split_line
tab_right
to_bottom
to_center
to_top
top_of_window
window_down
window_up
word_left
word_right

Cut, Copy, and Paste Text Editor Functions

Click an item below for more information.

copy
copy_block
copy_line
cut
cut_block
cut_line
paste
undo

Delete Text Editor Functions

Click an item below for more information.

backspace
cut
cut_line
delete
delete_line
delete_to_eol
delete_word_left
delete_word_right
replace
undo

File Text Editor Functions

Click an item below for more information.

close_window
compare
find_files_containing
list_files_containing
new_file
open_file
revert
save_file
save_file_close_window
save_all
save_all_exit
save_all_exit_windows
toggle_backup

Line Text Editor Functions

Click an item below for more information.

beginning_of_line
copy
copy_line
cursor_down
cursor_up
cut
cut_line
delete_line
delete_to_eol
end_of_line
enter
goto_line
select_line
select_line_down
select_line_up
select_to_bol
select_to_eol
split_line
to_bottom
to_center
to_top
window_down
window_up

Miscellaneous Text Editor Functions

Click an item below for more information.

Paragraph Text Editor Functions

Click an item below for more information.

document_preferences

enter

split_line

toggle_wordwrap

wrap_para

Text Selecting Text Editor Functions

Click an item below for more information.

- select_all
- select_char_left
- select_char_right
- select_line
- select_line_down
- select_line_up
- select_page_down
- select_page_up
- select_to_bol
- select_to_end
- select_to_eol
- select_to_top
- select_word
- select_word_left
- select_word_right
- unmark_block

The default keys are assigned so that pressing Shift along with a navigation key selects text as the cursor moves.

Search and Replace Text Editor Functions

Click an item below for more information.

- find
- find_again
- find_files_containing
- list_files_containing
- replace

Window Text Editor Functions

Click an item below for more information.

cascade

close_window

new_window

new_file

next_window

prev_window

restore_window

tile

to_bottom

to_center

to_top

window_down

window_up

zoom_window

Navigation Key Defaults

Key	Default Editor Function
←	cursor_left
Ctrl+←	word_left
→	cursor_right
Ctrl+→	word_right
↑	cursor_up
Ctrl+↑	window_down
↓	cursor_down
Ctrl+↓	window_up
PgUp	page_up
Ctrl+PgUp	top_of_window
PgDn	page_down
Ctrl+PgDn	bottom_of_window
Home	beginning_of_line
Ctrl+Home	beginning_of_buffer
End	end_of_line
Ctrl+End	end_of_buffer
Tab	tab_right
Enter	enter
Ctrl+G	goto_line

Text Selection Key Defaults

Key	Default Editor Function
Shift+↑	select_line_up
Shift+↓	select_line_down
Shift+←	select_char_left
Shift+Ctrl+	select_word_left
Shift+→	select_char_right
Shift+Ctrl+→	select_word_right
Shift+PgUp	select_page_up
Shift+PgDn	select_page_down
Shift+Home	select_to_bol
Shift+Ctrl+Home	select_to_top
Shift+End	select_to_eol
Shift+Ctrl+End	select_to_end
Esc	unmark_block

The default keys are assigned so that pressing Shift along with a navigation key selects text as the cursor moves.

F1...F12 Function Key Defaults

Key	Default Editor Function
F1	editor_help
F2	save_file
F3	open_file
F4	document_preferences
Ctrl+F4	close_window
F5	(none)
Ctrl+F5	restore_window
F6	(none)
Ctrl+F6	next_window
F7	record_macro
F8	play_macro
F9	(none)
F10	(none)
Ctrl+F10	zoom_window
F12	wrap_para

Other Key Defaults

Key	Default Editor Function
BkSp	backspace
Alt+BkSp	undo
Ctrl+BkSp	delete_word_left
Del	delete
Ctrl+Del	delete_word_right
Shift+Del	cut
Ins	toggle_insert
Ctrl+Ins	copy
Shift+Ins	paste
Enter	enter
numpad *	undo
numpad +	copy
numpad -	cut
Tab	tab_right
Alt+ -	close_window
Alt+D	delete_line
Alt+K	delete_to_eol
Alt+N	next_window
Alt+P	print
Alt+R	batch_help
Alt+V	about
Alt+X	save_all_exit

pcAnywhere Text Editor Procedures

Using the pcAnywhere Text Editor

The pcAnywhere Editor is the text editor supplied with pcAnywhere for Windows. It is the default editor used when creating and editing scripts.

Features of the pcAnywhere Editor include:

* A customizable keyboard--you can assign keystrokes to over 100 editing functions.
* MDI windows--you can open many edit windows, then tile, cascade, and minimize them. You can also open the same script file in more than one window.
* Uses as much memory as it needs for fast editing of large files.
* Flexible find and replace capabilities, including the use of regular expression text wildcards. The editor also remembers recently-used search strings.
* Ability to search for files containing a specific word or phrase and easily open them for inspection or editing.
* Customizable auto-save and automatic backup options.
* Ability to compare two files on a line-by-line basis.
* Auto-indent (handy for programmers).
* Supports word wrapping and paragraph reformatting for simple word processing tasks.
* Remembers recently-edited files--you can reopen the exact same set of files you used in your last editing session.

Opening files when the pcAnywhere Editor is running:

* Choose the editor's File Open command.
* Or, to open a file that you recently edited, choose it from the list at the bottom of the editor's File menu.
* Or, after using the editor's Find Files Containing command, use the List Found Files command to open any of the found files.

To automatically reload a set of files next time you start:

1. Use the Options Editor Preferences command and check the Restore Session check box.
2. Open the files you want to load the next time you start the editor.
3. Use the File Exit command to quit the editor (Note: don't close the files individually).
4. Next time you start the editor, it automatically opens the same files, arranging the windows in a cascading pattern.

Using Text Editor Macros

The pcAnywhere Editor allows you to record and play back a single keyboard macro. You can use this to automate simple, repetitive tasks.

To record an editor macro:

1. Press F7 (Alt+E,R) to start recording. The status line displays the word REC to indicate it is recording.
2. Type text and use menu commands and keystroke functions to accomplish the desired task.
3. Press F7 again (Alt+E,S) to stop recording. The status line displays the message Keyboard macro defined.

To play back the editor macro:

1. Make sure the desired window is active and the cursor is positioned correctly.
2. Press F8 (Alt+E,P) to begin replaying the macro.

Events that take place in dialog boxes are NOT recorded.

Up to 256 editing events can be recorded.

The current macro is not saved between editing sessions.

Using Multiple Edit Windows

The pcAnywhere Editor lets you see more than one document at a time and provides standard ways to access the documents.

To activate a window (bring it to the front):

Click any part of it, select it from the window list under the Window menu, or press Ctrl+F6 until it moves to the front.

To enlarge a window to see more text:

First maximize the application window (click the maximize button or press Alt+spacebar,X), then maximize the individual document window (click the maximize button or press Alt+hyphen, X).

To see two parts of a document simultaneously:

Activate the desired window. Use the Open Window command of the Window menu. You can scroll the windows independently.

To see as much as possible of two documents:

First maximize the application window, then choose the Window Tile command (Alt+W,T).

To see the title bars of all windows:

Choose the Window Cascade command (Alt+W,C).

To iconize a window:

Minimize the window into an icon (click the minimize button or press Alt+hyphen,N). It is now displayed at the bottom of the application window.

To restore an icon to a normal-size window:

Double-click the icon, or press Ctrl+F6 until it is highlighted, then press Alt+Hyphen,X to maximize it. You can

also select it from the Window menu.

To tidy up window icons:

Use the Window Arrange Icons command (Alt+W,A).

Searching with Regular Expressions

The Find, Replace, and Find Files Containing commands of the Search menu provide a powerful text wildcard search capability called regular expression searching.

Although not the full-blown "grep" utility known to UNIX programmers, this feature can be useful in many situations. For instance, it can be used for locating:

* Text in formatted layouts, such as phone numbers, zip codes, text in parentheses, quoted strings, etc.
* Programming elements and idioms; for instance, locating the start of any procedure.
* Proper names or the start of a sentence (i.e., first letter capitalized).
* Blank lines.
* A sequence of digits.
* Lines beginning or ending with some specified text.
* Alternate spellings of a word or name (e.g., Kathy or Cathy).
* Leading or trailing blanks in a line.

One of the powers of this feature over simple text-matching is that you can use it to select entire words or lines that contain a match. For instance,

goto	selects just that single word when found
<*goto*>	selects the entire line that contains the word

You can use this feature in editor macros to automate some operations. For instance, you can easily copy the matching line or word to the clipboard and paste it to a second file.

Examples

[KC]athy	Kathy or Cathy
[KC]athy [A-Z][~]@	Kathy or Cathy and her last name
[A-Z][a-z]@ [A-Z][~]@	any two-word proper name
[2-9]??-[0-9]???	any phone number
<*[2-9]??-[0-9]???*>	a line that contains any phone number
<* [A-Za-z]@(*){>	a line that defines a C function; e.g., int DoIt(x,y){
<{ finds "{" when at the start of a line	
<[\t][\t]>	any line containing only blanks or tabs
\\[nrt]	finds "\n" or "\r" or "\t"

Printing

The Desktop Editor provides printing capabilities that will more than satisfy your needs when you are working with text files, including font selection, and generation of page headers and footers. For more complex formatting, you may need to open the file using a full-featured word processor.

To print an entire document:

1. Activate the document you want to print by clicking its window title.
2. Deselect any selected text by pressing Esc.
3. Choose the File Printer Setup command to specify a printer.
4. Choose the File Page Setup command to select printing options. You can type text to be displayed at the bottom and/or top of each page, as well as select a font for the printout.
5. Choose the File Print command to send the file to the printer.

To print a partial document

1. Select the portion of the document you want to print.
2. Follow the steps above. You can modify the header or footer to indicate that this is a partial printout. All lines having any text selected are printed.

Comparing Two Files

The pcAnywhere Editor provides a handy file-comparison feature. This is especially useful for programmers who want to check for differences between versions of program source code.

To compare two files:

* (optional) Maximize the editor window and minimize all open documents. This provides screen space to show larger document windows when displayed.

1 Choose Compare from the File menu to display the Compare dialog box.

2 Select the two files you want to compare.

3 Leave the Line text box set at 1 for both files.

4 Choose the Horizontal option if you want the files to be displayed one above the other, or choose Vertical for a side-by-side display.

5 Click OK. The editor displays both files, highlighting the first difference it finds. If no differences are found, it displays a message on the status line.

6 The editor displays a small dialog box. Click Find Next... to advance through the files, or click Cancel when satisfied.

Finding Text in an Open File

The pcAnywhere Editor can search for simple text strings or for special text wildcards. See Searching with Regular Expressions for a description of the latter capability. See Searching for Files for a way to locate all files which contain a specified text string.

To search an entire file:

1. Open and activate the file you want to search.
2. Press Ctrl+Home to get to the top of the document (the search always starts at the current cursor location).
3. Press Ctrl+S or use the Find command of the Search menu to bring up the Find dialog box.
4. Enter the text you want to find into the Pattern text box.

   Or, press Alt+{bmp symDnAro.Bmp} to choose from a list of recently-used search strings.

   Or, you can select text before opening the dialog and the selected text is placed into the text box automatically.
5. Uncheck the Match Upper/Lowercase check box to provide the best chance of finding a similar match. Leave the Regular Expressions check box unchecked to speed up the search.
6. Click Next to start searching. If any match is found, it is displayed and highlighted. Otherwise, the message Pattern not found is displayed in the status line.

To continue a search:

* Press Ctrl+A or use the Find Again command on the Search menu. If another match is found, it is displayed and highlighted. Otherwise, the message Pattern not found is displayed in the status line.

To search backward through a document:

1. Position the cursor where you want the search to start.
2. Press Ctrl+S, and follow the steps described above, except use the Previous button in the dialog box to begin the search.
3. Pressing Ctrl+A continues the search backward through the document until it hits the top of the file.

Finding Files that Contain Specified Text

The pcAnywhere Editor can search the disk for files containing a specified string of text. This is particularly useful for programmers wishing to inspect or modify all source code files that use a particular function or variable.

To find files containing a search string:

1. Use the Find Files Containing command on the Search menu to bring up the Find Files Containing dialog box.
2. In the Pattern text box, enter the search string (it can contain regular expression wildcard characters).

   Or, select from the list of previously-used search strings.
3. In the Files text box, enter one or more wildcard file specifications separated by spaces.

   Or, select from the list of previously-used filespecs.

   Or, click Directory... and select the directory from the tree display.
4. Clear the Match Upper/Lowercase check box, unless you want the search to be case-sensitive.
5. Check the Regular Expression check box only if you have used text wildcards in the Pattern text box.
6. Click OK to start the search.
7. If any files containing the specified string are found, a list of filenames appears. Just double-click a filename to open it. The first occurrence of the found text is displayed.

To inspect or edit other files that contain the found string:

1. Use the List Found Files command on the Search menu to display the List Found Files dialog box.
2. Select a file from the scrolling list and click Open.
3. As you open files from this dialog, they are removed from the list.

Customizing Text Editor Keystrokes

The pcAnywhere Editor lets you customize the keyboard; you can assign any keystroke to any of over 100 editor functions.

Most users are satisfied with the default key assignments. However, when you are accustomed to another editor, you may find it convenient to change some of the pcAnywhere Editor keystrokes to match your old habits (so your fingers can do the thinking...).

To add a keystroke to a function:

1. Browse the Editor Functions: Functional Groups help topic to familiarize yourself with the available functions.
2. Use the Key Assignments command on the Options menu to display the Key Assignments dialog box.
3. Press ↓ and scroll through the Functions list box until you highlight the function you want to access; its name appears in the box on top.

   The Current Keys box at the bottom displays what key or keys are currently assigned to the function. You are about to add another keystroke to those assigned to the function.
4. Press Tab and scroll through the Keys list box until you find the keystroke you want to use to access that function.

   The Current Function box at the bottom of the dialog box displays what function (if any) is currently assigned to that keystroke. If you proceed, you lose that assignment, since a key can be assigned to only one function.
5. Click Assign.
6. Click Save to save the changes to the file, DEFAULT.KEY.
7. Click OK.

To change a keystroke assignment:

1. Use the Key Assignments command on the Options menu to display the Key Assignments dialog box.
2. Press Tab and ↓ and scroll through the Keys list box until you find the keystroke you want to change.

   The Current Function box at the bottom displays what function (if any) is currently assigned to that keystroke. If you proceed, you lose that assignment, since a key can be assigned to only one function.
3. Press Shift+Tab and ↓ and scroll through the Functions list box until you find the editor function you want to map to the selected keystroke.
4. Click Assign.
5. Click Save to make the assignment permanent.
6. Click OK.

Finding Which Key Performs a Text Editor Function

The pcAnywhere Text Editor features over one hundred editing functions. Click below to see more on these functions:

Editor Functions: Functional Groups

Since some keystrokes may have been reassigned (or if you want avoid this help system), you can find the current key assignments as follows:

To see what key performs a function:

1. Use the Key Assignments command on the Options menu to bring up the Key Assignments dialog box.
2. Press ↓ and scroll through the Functions list box until you highlight the function you want to access; its name appears in the box on top.

   The Current Keys box at the bottom displays what key or keys are currently assigned to the function.
3. Click Cancel to avoid changing any settings.

Customizing the pcAnywhere Text Editor

The pcAnywhere Editor lets you customize many of its features. If you are already familiar with another text editor, you can modify the pcAnywhere Editor to conform to your editing habits. And, as you become comfortable with the pcAnywhere Editor, you can make minor adjustments to speed up common operations.

Some customization is automatic. For example, the pcAnywhere Editor automatically remembers:

* Files that you have recently edited (they are placed on the File menu).
* Search strings you have previously used.
* File types recently used when finding files.
* Files that you have recently compared.
* Page printing settings, including margins, font face and size, headers, and footers.

Here are some other features and the menu commands that let you customize them:

Feature	Menu command
Automatic file backups	Options Editor Preferences
Automatic file saving	Options Editor Preferences
Automatic line indenting	Options Document Preferences
Automatic file loading	Options Editor Preferences
Convert tabs to spaces	Options Document Preferences
Cursor shape	Options Editor Preferences
Features not on menus	Options Key Assignments
File locking	Options Editor Preferences
Header/Footer text	Options Document Preferences
Keyboard macro	Edit Record Macro (see Using Macros)
Keystroke assignments	Options Key Assignments
Line indenting	Options Document Preferences
Printer Font	Options Document Preferences
Printer Margins	Options Document Preferences
Programmer's indenting	Options Document Preferences
Reload recent files	Options Editor Preferences
Remove trailing spaces	Options Editor Preferences
Screen Font	Options Editor Preferences
Screen margin settings	Options Document Preferences
Tab settings	Options Document Preferences
Typing replaces selected text	Options Editor Preferences
Undo levels	Options Editor Preferences
Word wrapping	Options Document Preferences

word

For the purposes of word-wrap:

> An unbroken sequence of non-blank characters.

As defined for cursor motion, selection, and deletion:

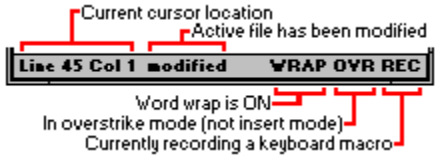
> A sequence of characters unbroken by space, tab, or any common programing text delimiters such as ; , { } ( ), etc.

paragraph

A sequence of lines of text, containing no blank lines.

When determining how to wrap a paragraph, the editor looks backward from the cursor location until it finds a blank line, then assumes that the paragraph includes all lines after that, down to, but not including, the first blank line.

status line



The pcAnywhere Editor status line, at the bottom of its application window, displays information about the currently active edit window.

Information, including error messages and search-and-replace status is displayed momentarily on the left side of the status line, until the next keystroke or mouse click.

delimiter

When using editor functions to advance or select by "words," the following characters are considered as word-ending delimiters:

tab (0x09)

space (0x20)

end-of-line

_ ( ) " ' , # $ @ ! % ^ & * { } [ ] \ ? / | < > ; . + - = ~ :

regular expression "wildcard" codes

? any single character or space

* zero or more characters or spaces

[chars] any one of the characters between [and]

[**~**chars] any one character except those after the **~**

[c1-c2] any one of the range of characters c1 to c2

@ zero or more of the previous character

% or < the beginning of a line

$ or > the end of a line

\t a tab character (0x08)

\f a formfeed character (0x0c)

\c the literal character c; e.g. \\ means \

wildcard file specification

You can use wildcard characters to specify a set of files which share similar names. The wildcard characters are:

* any 1-to-8 character filename (or 0-to-3 character file extension)

? any single character

Examples:

. any filename and any extension

*.bak any filename with an extension of .BAK

*.? any file that has a 1-character extension

*. any file that has no extension

*.WK? any filename that has an extension that starts with .WK and then any other single character

rpt??jul.* any filename starting with RPT, then any two characters, then JUL, and having any extension

Using pcAnywhere scripts

pcAnywhere includes a script language that allows you to:

Perform operations automatically when calling online services.

Automate some procedures during remote control sessions.

Use scripts to automate:

- Running programs
- Transferring files
- Performing arithmetic operations
- Other procedures

To create a script:

Use the Script Editor. For more information on the Editor, refer to the pcAnywhere Editor Help Contents.

Record the activities of a session with an online service to a script file, so that the procedures in that session can be repeated by executing the script in a later session. To do this, check the Begin Script Recording After Connection option in the Session tab of an online service connection item.

To execute a script upon connection in a remote control session:

1 Check **Run Upon Connection** in the Automated Tasks tab of a remote control connection item! PI(`Winaw32.hlp',`Bmap_REmcon_connection_item').

2 Click Script and type the name of the script file you want to run or click Browse to locate the script file.

To execute a script upon connection in an online service session:

1 Check **Run Script File After Connect** in the Session tab of an online service connection item.

2 Type the name of the script file you want to run or click Browse to locate the script file.

To run a script during a remote control session:

- Choose Scripts from the pcAnywhere remote menu, then select the script you want to execute.

To run a script during an online service session:

1 Choose Scripts from the File menu of the Terminal window.

2 Select the script you want to run.

3 Click Run.

The pcAnywhere Script Editor: Help Contents

The pcAnywhere Editor Menu Commands

File menu

New

Open

Close

Save

Save As

Save All

Insert

Revert

Write Block

Page Setup

Print

Compare

Exit

(list of files)

Edit menu

Undo

Cut

Copy

Paste

Delete

Select All

Time/Date

Wrap Paragraph

Word Wrap

Record Macro

Playback Macro

Search menu

Find

Find Again

Replace

Find Files Containing

List Found Files

Goto Line

Options menu

Document Preferences

Editor Preferences

Key Assignments

Tool menu

Reference

Window menu

New Window

Cascade

Tile

Arrange Icons

(list of windows)

See Also

Using the pcAnywhere Editor
Text Editor Functions

The File menu

Commands in this menu let you create, edit, and print script files as well as providing access to other file-related functions.

New	Create a new script file.
Open	Open an existing script file.
Close	Close the active editing window.
Save	Save file in the active editing window.
Save As	Save file in active editing window using different name.
Save All	Save all open files.
Insert	Insert contents of a file at cursor position.
Revert	Undo all changes since last save.
Write Block	Write selected block to a file.
Page Setup	Set headers, footers, margins, etc.
Print	Print current script file.
Compare	Compare contents of two files.
ExitClose the pcAnywhere Editor windows and exit.	
(list of files)	Open or activate a recently used file.

New command (File menu)

Function name:**new_file**

Default keys:**Alt+F,N**

This command opens a new untitled editing window. Use the Save or Save As… command to assign a name to the file.

Open command (File menu)

Function name:**open_file**
Default keys: **F3**

This command displays an Open File dialog box. Use it to select a file to edit.

Open File dialog box (Standard Browse)

Use this dialog box to open an existing script file for editing.

File Name: Type or select the filename you want to open. This box lists files with the extension selected in the List Files of Type box.

List Files of Type: Select the type of file you want to open

Script Files (*.SCR): Lists all files in the current folder that were saved with an *.SCR extension.

Drives: Select the drive on which the file you want to open is stored.

Folders: Select the folder in which the file you want to open is stored.

Close command (File menu)

Function name:**close_window**
Default keys:**Alt+F,C**

This command closes the active editing window.

If you have modified the file and not yet saved your changes, (**and** it is not open in any other editing window), then the pcAnywhere Editor prompts you to save your changes before it closes the file.

Save command (File menu)

Function name:**save_file**
Default keys: **F2**

This command immediately saves the file in the active window.

If the file does not yet have a name, a Save As... dialog box appears and prompts you for a path and filename.

Save As dialog box (Standard Browse)

Use this dialog box to specify a path and filename under which to save your script file.

File Name: Type or select the filename under which you want to save your script. This box lists files with the extension selected in the List Files of Type box.

List Files of Type: Select the type of file to save. If you are saving a script file, be sure to specify an extension of .SCR.

Script Files (*.SCR): Lists all files in the current folder that were saved with an *.SCR extension.

Drives: Select the drive on which you want to store your script file.

Folders: Select the folder in which you want to store your script file.

Save As command (File menu)

Function name:(none)

Default keys:**Alt+F,A**

This command displays a Save As... dialog box. Use it to select a new path and filename for the file displayed in the active editing window.

You can use this command to save the file with a new name; for example, make a backup of your file by saving it under a different name.

Save All command (File menu)

Function name:**save_all**
Default keys:**Alt+F,L**

This command saves all open files.

If any windows are untitled, a Save As... dialog box appears. Use this to specify a path and filename for any open, untitled files.

Insert command (File menu)

Function name:(none)
Default keys:**Alt+F,I**

Use this command to insert the contents of one file into another. The file is inserted at the current cursor location.

An Insert File dialog box appears and prompts you for the name of the file to insert.

Insert File dialog box (Standard Browse)

Use this dialog box to specify the file to be inserted at the current cursor position.

File Name: Type or select the name of the file you want to insert into the active editing window.

List Files of Type: Select the type of files to display. Script source files have an extension of .SCR.

Script Files (*.SCR): Lists all script files in the current folder.

Drives: Select the drive on which the desired file is stored.

Folders: Select the folder in which the desired file is stored.

Revert command (File menu)

Function name:**revert**

Default keys:**Alt+F,V**

This command undoes all editing changes since the file was last saved. Before reading the file from disk, the pcAnywhere Editor displays a confirmation box, allowing you to cancel the operation.

Write Block command (File menu)

Function name:(none)
Default keys:**Alt+F,W**

This command copies the currently selected text to a file. It is enabled only when text is selected.

When the Write Block... dialog box appears, specify a path and filename under which to save the text.

Write Block dialog box (Standard Browse)

Use this dialog box to specify the path and filename under which to store the selected text.

File Name: Type or select the name of the file under which you want to save the selected text.

List Files of Type: Select the type of files to display. Script source files have an extension of .SCR.

Script Files (*.SCR): Lists all script files in the current folder.

Drives: Select the drive on which to store the new file.

Folders: Select the folder in which to store the new file.

Page Setup command (File menu)

Function name:(none)

Default keys:**Alt+F,G**

This command displays a Page Setup dialog box which lets you select a paper size, font, margins, and header and footer text to be used when files are printed. You can also select a printer.

Use the File Print command to print the document.

Page Setup dialog box

Use this dialog box to set page formatting options, preview a print job, and select a printer. These settings take effect the next time you use the Print command.

Preview area

Displays thumbnail view of print job with options (below) as currently selected.

Paper Size and **Source**

Choose paper size and source (tray, manual feed, and so on).

Header and **Footer**:

Enter one line of text to be centered at the top (a header) and/or bottom (a footer) of each page. Or, click the prompt button (or press Alt+↓) to choose from a list of previously-used settings.

You can embed the following special-purpose codes in the text:

%f	Full path and filename of active document
%d	Current date and time
%p	Current page number

To omit a header or footer, leave its text box empty.

In addition, choose a previous header or footer code from the list of options available from the drop-down list.

Margins:

Set the print margins. The units of measure are the standard units used in your country (inches or centimeters), as set in the Windows Control Panel.

Font:

Click this to display a Printer Font dialog box which lets you choose a typeface and size for the printed output. The entire document, including headers and footers, uses the selected font. The specified settings are used for the current session and all subsequent sessions, until changed.

See Also

Printing a Document
Printing a Block of Text
The File Print command
Printer Font dialog box

Printer Font dialog box

Use this dialog box to choose a typeface and size for your printed output. The Printer Font dialog box appears when you click the Font... button from the Page Setup dialog box.

Font:

Choose from the list of available fonts. Only fonts available for the current printer are shown (use the Printer Setup command on the File menu to choose a different printer and the Windows Control Panel to install additional fonts).

Size:

Select from the list of available sizes for the currently-selected font.

See Also

Printing a Document
The Page Setup command
The File Print command

Print command (File menu)

Function name:**print**
Default keys:**Alt+P** or **Alt+F,P**

If text is selected in the active window, this command sends the selected text to the default printer.

If no text is selected, the entire file is printed.

Use the Page Setup… command to select a printer and specify print options before using this command.

Compare command (File menu)

Function name:**compare**

Default keys:**Alt+F,E**

This command displays a Compare dialog box which lets you select two files to compare.

If either file is currently open, the pcAnywhere Editor uses the version which is in memory, rather than reading the file from the disk. If you want to pinpoint recent changes, first save the open file under a different name (use the Save As... command), then compare it to the file on disk.

The pcAnywhere Editor performs the comparison on a line-by-line basis. Both files are displayed and the first mismatch is highlighted.

Compare dialog box

Use this dialog box to choose two files to compare, their starting line numbers, and how to display the ouput. The Compare dialog box appears when you choose Compare from the File menu.

File 1 and **File 2**:

Enter the names of the files to compare.

Or, click the prompt button (or press Alt+↓) to choose from a list of previously-used files.
Or, click Browse... to locate the desired files.

Line:

To compare all lines in both files, use the default setting of 1. Otherwise, specify a starting line number for each file.

Display:

Select an option button:

* Horizontal: The files are displayed one above the other.
* Vertical: The files are displayed side-by-side.

Browse:

Click this button to display a Compare dialog box which helps you locate a desired file. The selected file is displayed in the File text box which is active.

See Also

Comparing Two Files
The File Compare command

Exit command (File menu)

Function name:**exit**
Default keys:**Alt+F4** or **Alt+F,X**

This command closes all pcAnywhere Editor windows and exits.

The pcAnywhere Editor prompts you to save changes you have made to any open file.

List of files (1 to 4) (File menu)

Function name:(none)

Default keys:**Alt+F,1...4**

This command lets you quickly open a recently edited file. The last four files edited appear at the bottom of the File menu. To open one of these files, press Alt+F then 1, 2, 3, or 4.

If the selected file is currently in memory, its window is brought to the front. You can also activate any open or minimized window by selecting it from the Window menu.

The Edit menu

Commands in this menu provide text editing tools,access to the clipboard, and keyboard macro control.

Undo	Undo the most recent text-editing change.
Cut	Copy the current selection to the clipboard and then delete that text.
Copy	Copy selected text to the clipboard.
Paste	Paste clipboard contents to the cursor position.
Delete	Delete current selection.
Select All	Select all text in the active file.
Time/Date	Insert the current date and time at the cursor.
Wrap Paragraph	Reformat the paragraph at the cursor, wrapping at the current margin settings.
Word Wrap	Toggle automatic paragraph wrapping on or off.
Record Macro	(toggles to Stop Recording Macro) Record subsequent keystrokes for automatic recall.
Playback Macro	Replay the keystrokes saved during the most recent Record Macro command.

By default, the pcAnywhere Editor uses standard keystrokes for text selection and clipboard operations. However, you can assign different (perhaps more convenient) keystrokes to these functions. See Customizing Editor Keystrokes for details.

Undo command (Edit menu)

Function name:**undo**

Default keys:**Alt+BkSp** or **Gray ***

Reverses the effects of a previous editing operation. Each Undo reverses the effects of one editor function or keystroke. The number of operations that can be undone is specified in the Editor Preferences dialog box (from the Options Menu).

Only changes made since a file was last saved can be undone.

Cut command (Edit menu)

Function name:**cut**
Default keys:**Shift+Del** or **Gray -**

Deletes the selected text from the active window and moves it to the clipboard.

If no text is selected, and if the Cut/Copy Current Line if No Text is Selected check box in the Editor Preferences dialog box is checked, the current line is cut.

Use the Copy command (on the Edit menu) if you want to copy the selection to the clipboard without deleting it. Use Delete if you want to discard the line without changing the contents of the clipboard.

Copy command (Edit menu)

Function name:**copy**
Default keys:**Ctrl+Insert** or **Gray +**

Copies the selected text from the active window into the clipboard.

If no text is selected, and if the Cut/Copy Current Line if No Text is Selected check box in the Editor Preferences dialog box is checked, the current line is copied.

Paste command (Edit menu)

Function name:**paste**
Default key:**Shift+Insert**

Inserts the contents of the clipboard at the current cursor position.

Delete command (Edit menu)

Function name:**delete**
Default key: **Del**

Deletes the currently selected text. If no text is selected, deletes the character to the right of the cursor.

Use the Cut command (on the Edit menu) if you want to copy the selection to the clipboard and simultaneously delete it from your file.

Select All command (Edit menu)

Function name:**select_all**
Default key:**Alt+E,A**

Selects all text in the active window. The cursor is moved to the end of the file.

This is the same as pressing Ctrl+Home and then Shift+Ctrl+End.

Time/Date command (Edit menu)

Function name:**stamp**
Default key:**Shift+F2**

Inserts the date and time at the cursor location.

Word Wrap command (Edit menu)

Function name:**toggle_wordwrap**

Default key:**Ctrl+W**

Enables or disables word wrap in the active window. Word wrap is used when writing narrative paragraphs; turn this feature off when writing program source code or creating formatted tables.

When word wrap is enabled, typing beyond the right margin causes the current word to be moved down to the next line. The word WRAP appears on the status line.

The right margin setting can be changed from the Document Preferences dialog box dialog box on the Options menu.

Use the Wrap Paragraph command on the Edit menu to reformat a paragraph and wrap all lines at the margin.

The pcAnywhere Editor displays text using a monospaced font, so word wrapping is based on a uniform character width. If you use a proportional-spaced font when you print the document, the right edge appears more ragged in the printout than it does on the screen.

Wrap Paragraph command (Edit menu)

Function name:**wrap_para**

Default key: **F12**

Reformats the current paragraph (or the previous paragraph, if the cursor is between paragraphs) so that all the words fit within the margins. The cursor moves to the end of the paragraph.

This command does nothing if word wrap is turned off. Use the Word Wrap command (on the Edit menu) to enable this feature.

Word wrap is used when writing narrative paragraphs. Don't use this feature when writing program source code or creating formatted tables.

Change the margin settings from the Document Preferences dialog box. To display this dialog box, choose Document Preferences from the Options menu.

Record Macro command (Edit menu)

Function name:**record_macro**

Default key: **F7**

Begins or ends a keyboard macro definition. While recording a macro, REC appears on the status line, and the menu command toggles to Stop Recording Macro. All keystrokes and functions are recorded until you choose Stop Recording Macro from the Edit menu.

You can use the Playback Macro command to replay the recorded sequence.

See Using Keyboard Macros for related details.

Play Back Macro command (Edit menu)

Function name:**play_macro**
Default key: **F8**

Replays the sequence of keystrokes and functions saved by the Record Macro command.

See Using Keyboard Macros for related details.

The Search Menu

Commands in this menu let you search for text in the current file or in a set of files on disk. They also provide a quick way to move to a line given its line number.

Find Search for text or a regular expression.

Find Again	Continue search started with the Find command.
Replace	Search for and replace text or regular expression.
Find Files Containing	Generate a list of files which contain the specified text or regular expression.
List Found Files	See a list of files found; choose one to edit.
Goto Line	Move the cursor to a line specified by its line number.

Find command (Search menu)

Function name:**find**

Default key:**Ctrl+S**

This command searches the active document for specified text or a regular expression. It displays the Find dialog box which lets you specify search criteria such as what text to find and in which direction to perform the search.

If you select text before choosing this command, the selected text appears as the default search string in the dialog box.

The search commences at the current cursor location of the active document. If the search is successful, the matcing text is highlighted. If it fails, the status line displays the message Pattern not found.

- The pcAnywhere Editor cannot find text that spans two lines.

Find dialog box

This dialog box appears when you choose the Search Find command. Use it to specify the text you want to find and in which direction to perform the search.

Pattern:

Enter the text you want to find, or click the prompt button (or press Alt+↓) to choose from a list of previously-used search strings.

If you want to use regular expression wildcards in your search string, check the Regular Expression check box.

By default, the pcAnywhere Editor uses any currently selected text as the search string.

The pcAnywhere Editor cannot find text that spans two lines.

Match Upper/Lowercase:

When this check box is checked, the pcAnywhere Editor checks the case of the text in the Pattern combination box and returns only exact matches.

When cleared, the editor considers text a match even when the case is not the same. For example, goto matches GOTO and Goto.

Regular Expression:

Check this if you want to enable the powerful and flexible regular expression pattern matching. Leave this unchecked (for speed) if you simply want to locate a string.

Next:

Click this or press Enter to confirm your settings and begin searching forward from the cursor location. If a match is found, the found text is highlighted.

Previous:

Click this to confirm your settings and begin searching backward from the cursor location.

See Also

Searching with Regular Expressions

Finding Files that Contain Specified Text

Search Functions

Find Again command (Search menu)

Function name:**find_again**

Default key:**Ctrl+A**

This command continues the search begun by the Search Find command. The search resumes from the current cursor position in the active document and proceeds in the direction previously specified.

If the search is successful, the matching text is highlighted. If it fails, the status line displays the message Pattern not found.

Replace command (Search menu)

Function name:**replace**

Default key:**Ctrl+R**

This command displays a Replace dialog box which lets you find and replace occurrences of text with different text. You can also perform a search and replace on a regular expression.

If you select some text before using this command, that text appears as the default search string in the dialog box.

The search commences at the current cursor location of the active document. If the search succeeds (and if you checked the Confirm Changes check box), the matching text is highlighted and you can choose to replace or skip each occurrence. The status line reports how many changes were made.

If the search fails, the status line displays the message Pattern not found.

- The pcAnywhere Editor cannot find text that spans two lines.

Replace dialog box

Use this dialog box to specify the text to find and the text which replaces it. The search and replace operation always proceeds forward from the current cursor location in the active window.

Search For:

Enter the text you want to find or click the prompt button (or press Alt+↓) to choose from a list of previously-used search strings.

If you want to use regular expression wildcards in your search string, check the Regular Expression check box.

- By default, the pcAnywhere Editor uses any currently selected text as the search string.

The pcAnywhere Editor cannot find text that spans two lines.

Replace With:

Enter the text you want to use as the replacement string or click the prompt button (or press Alt+↓) to choose from a list of previously-used replacement strings. You cannot use a text wildcard in this text box.

If you leave this field blank, each occurrence of the search string is deleted (that is, replaced with nothing).

Match Upper/Lowercase:

When this check box is checked, the pcAnywhere Editor checks the case of the text in the Pattern combination box and returns only exact matches.

When cleared, the editor considers text a match even when the case is not the same. For example, goto matches GOTO and Goto.

Regular Expression:

Check this if you want to enable the powerful and flexible regular expression pattern matching. Leave this unchecked (for speed) if you simply want to locate a string.

Confirm Changes:

Check this if you want to be prompted before a change is performed. When unchecked, the editor replaces all occurrences of the Search For text, starting at the current cursor location and proceeding to the end of the file, without prompting you.

It is a good idea to leave this checked and confirm the first one or two changes. When you're satisfied that an appropriate string has been specified in the Search For text box, you can uncheck this checkbox let the editor quickly finish the job.

Repeated use of the Undo command on the Edit menu can repair the effects of undesired changes.

If Confirm Changes is checked, you are prompted to accept or skip each occurrence of the found text. Otherwise, the editor automatically replaces all occurrences from the cursor location onward.

The number of changes made appears on the status line.

See Also

Searching with Regular Expressions
Finding Files that Contain Specific Text
Search Functions

Find Files Containing command (Search menu)

Function name:**find_files_containing**

Default key:**Ctrl+F**

This command displays the Find Files Containing dialog box, which lets you search the disk for files which contain specified text or a regular expression. Specify the text to search for, the types of files to search and a folder.

If one or more files are found to contain the search string, the List Found Files dialog box appears. You can open any of the files from here and edit it.

After opening a found file, the first occurrence of the search text is displayed and highlighted. Use Ctrl+A to find other occurrences.

Find Files Containing dialog box

Use this dialog box to choose what type(s) of files to scan and specify the text you want to find. The Find Files Containing dialog box appears when you choose Find Files Containing… from the Search menu.

Pattern:

Enter the text you want to find or click the prompt button (or press Alt+) to choose from a list of previously-used search strings.

If you want to use regular expression wildcards in your search string, check the Regular Expression check box.

The pcAnywhere Editor cannot find text that spans two lines.

Files:

Enter one or more wildcards separated by spaces. Or, click the prompt button (or press Alt+) to choose from a list of previously-used file specifications.

You can include a drive ID and path in any file specification. If omitted, only files in the current default folder are searched.

Examples:

*.c *.h *.asm	find a variable or function name in several types of program source code files
c:\wp\ltrs*.doc c:\wp\old*.*	look at files in two directories

Match Upper/Lowercase:

When this check box is checked, the pcAnywhere Editor checks the case of the text in the Pattern combination box and returns only exact matches.

When unchecked, the editor considers text a match even when the case is not the same. For example, goto matches GOTO and Goto.

Regular Expression:

Check this if you want to enable the powerful and flexible regular expression pattern matching. Leave this unchecked (for speed) if you simply want to locate a string.

See Also

Finding Files that Contain Specified Text

Searching with Regular Expressions

The List Found Files command

Search Functions

List Found Files command (Search menu)

Function name:**list_files_containing**
Default key:**Ctrl+L**

This command displays a List Found Files dialog box containing the the files located using the Search Find Files Containing command. Use it to examine each file.

As you open each file from this dialog box, it is removed from the list. This makes it easy to process the files one-by-one, examining and/or making changes to each file.

After opening a found file, the first occurrence of the search text is displayed and highlighted. Use Ctrl+A to find other occurrences.

List Found Files dialog box

This dialog box appears when a search started by the Search Find Files Containing command locates one or more files. Thereafter, you can use the List Found Files command to view the list and choose files to open.

Pattern and **Filter**:

These are displayed to remind you of the search criteria used to obtain the files in this list.

Files:

This lists the files found by the most recent use of the Find Files Containing command (on the Search menu), less any files that have already been opened for editing.

Scroll through the list to select a file. Click Open to open the file and remove it from the list.

Open:

This opens the selected file for editing. On future uses of the List Found Files command, the opened file does not appear in the list.

See Also

Finding Files that Contain Specified Text
Searching with Regular Expressions
The List Found Files command
Search Functions

Goto Line command (Search menu)

Function name:**goto_line**
Default key:**Ctrl+G**

This command displays a Goto Line dialog box which lets you specify the number of a line to jump to.

The active document is scrolled to the requested line. If any text is currently selected, the selection is extended to include that line.

This command is particularly useful for programmers. When a compiler reports an error, you can quickly move to the line containing the error.

Goto Line dialog box

Use this dialog box to specify the line number at which to position the cursor.

Line Number:

Enter a number between 0 and 32,365.

The file in the active window is scrolled to the requested line. If any text is currently selected, the selection is extended to include that line.

See Also

The Goto Line command
Cursor Motion Functions
Search Functions

The Options menu

Commands in this menu set editor and document preferences.

Document Preferences	Set margin, tabs, indent, etc. for current file only or defaults for all files.
Editor Preferences	Set screen font, cursor, backup, autosave.
Key Assignments	Customize keystrokes for fast access to editor functions.

Document Preferences command (Options menu)

Function name:**document_preferences**

Default key: **F4**

This command displays a Document Preferences dialog box which lets you select options for the currently active document.

Document preference settings are used for the current document for the current editing session. They are not saved when you save your script file. However, you can specify default settings, to be used in all editing sessions, from the Editor Preferences dialog box.

See Also

Customizing the pcAnywhere Editor
Document Preferences dialog box
The Editor Preferences command
The Key Assignments command

Document Preferences dialog box

Use this dialog box to customize settings for the current active document. This dialog box appears when you choose Document Preferences from the Options menu.

These settings are used for the current document during the current editing session. They are not saved with the document. However, you can set default preferences by choosing Editor Preferences from the Options menu.

Tab Spacing:

Sets your desired tab-stops for the current document. When you press the Tab key, the cursor advances this many character-widths.

Right Margin:

Sets the wrapping-point to be used when word wrap is in effect. Enter a number of columns. The right margin is ignored unless the Word Wrap checkbox is checked (see below).

To set the margins for printing, choose Page Setup from the File menu.

Word Wrap:

Check this to enable word-wrap in the current document for the current editing session. You can toggle this setting by using the Word Wrap command on the Edit menu.

* When checked, typing beyond the right margin causes the current word to be moved down to the next lower line. Also, this enables the Wrap Paragraph command of the Edit menu.
* When cleared, the right margin setting is ignored.

Auto Indent:

When checked, pressing Enter positions the cursor directly below the first non-blank character in the previous line. This is handy for programmers.

Expand Tabs with Spaces:

When checked, the editor replaces TAB characters (0x09) with a series of spaces (0x20) as it reads a file. It uses the tab settings set above. When it saves the file, all TAB characters are gone.

Save as Default Settings:

When checked, the above settings are used as the defaults for files opened in the future. Otherwise, the settings are used only for the current file during the current session.

See Also

Customizing the pcAnywhere Editor
The Editor Preferences command
The Document Preferences command
The Key Assignments command

Editor Preferences command (Options menu)

Function name:(none)

Default key:**Alt+O,E**

This command displays an Editor Preferences dialog box which lets you select options to be used by the editor for all documents. These include the display font, cursor style, backup, autosave, and undo level preferences.

Editor preference settings are saved and used in the current editing session and remembered for use in future sessions.

See Also

Customizing the pcAnywhere Editor
Editor Preferences dialog box
The Page Setup command
The Editor Preferences command
The Key Assignments command

Editor Preferences dialog box

Use this dialog box to customize some of the pcAnywhere Editor features. This dialog box appears when you choose Editor Preferences from the Options menu.

Changes made here are saved automatically and used each time the editor starts.

Font:

Select one of the option buttons. The font and size that is displayed depends on settings in your SYSTEM.INI file in the [boot] section:

* System Fixed Font depends on **fixedfon.fon=**
* ANSI Fixed Font depends on **fonts.fon=**
* OEM Fixed Font depends on **oemfonts.fon=**

The OEM font contains the IBM text-mode box-and-line drawing characters.

Cursor:

Select the option button corresponding to how you want the text cursor to be displayed. Check the Blinking check box to enable a blinking cursor.

* Block Covers the current character.
* Underline Underlines the current character.
* Vertical Bar Standard "insertion point" caret, displayed between characters.
* Blinking The cursor blinks when this is checked. You can set the blink rate from the Windows Control Panel.

Autosave:

If you want the editor to automatically save open files, put non-zero values in either or both of the text boxes.

Undo Levels:

Sets how many edit changes are remembered for the undo function. A large number tends to use up memory. Setting this to 0 disables the undo function.

Restore Session:

When checked, each file that was open last time you exited the editor is opened automatically next time you start the editor.

Typing Replaces Selection:

When checked, the editor follows the Windows standard convention of replacing any selected text with the first character typed or pasted. When unchecked, typing or pasting inserts the text to the left of the current selection.

Make Backup Files:

When checked, each time you save a file, the editor makes a copy of the original, giving it the same name, but with an extension of .BAK.

File Locking:

When checked, the editor locks each file when you open it and unlocks it only when you close its window. When a file is locked, no other program can delete or modify it. It only has an effect if a network is active and/or the program SHARE.EXE has been executed.

Cut/Copy Current Line if No Text is Selected:

When checked, you can quickly cut or copy the current line using the standard cut or copy keystrokes (without having to first select the line). When unchecked, you can cut or copy a line by using the keys assigned to the cut_line or copy_line functions.

Remove Trailing Spaces:

When checked, the editor discards spaces and tabs characters that are at the end of each line.

See Also

Customizing the pcAnywhere Editor
The Editor Preferences command
The Document Preferences command
The Key Assignments command

Key Assignments command (Options menu)

Function name:(none)

Default key:**Alt+O,K**

This command displays a Key Assignments dialog box which lets you customize the way the editor interprets and acts on keystrokes. You can assign your own favorite shortcut keys to any editor function.

You can also use this command to see what keystrokes are currently mapped to the editor functions. See also:

Editor Functions: Functional Groups

Be sure to use the Save button in the dialog box if you want your key assignments to be remembered for future editing sessions.

See Also

Customizing the pcAnywhere Editor

Key Assignments dialog box

The Document Preferences command

The Editor Preferences command

Key Assignments dialog box

Use this dialog box to associate custom keystrokes with editor functions. You can also use it to look up the currently-assigned keystroke for an editor function. This dialog box appears when you choose Key Assignments from the Options menu.

In order to make a keystroke assignment permanent, you must select a function and a key, click Assign, then click Save.

Function:

Select a function from this list box. As you move the cursor through the items in the list box, the current keystroke for that function is displayed in the Current Keys box below. See also:

Editor Functions: Functional Groups

Key:

Select a keystroke from the list box. Only the listed keystrokes can be used.

Assign and **Unassign**:

When a name is in both the Function and Key text boxes, you can click these buttons to associate a keystroke with a function or dissolve the association.

You can assign more than one keystroke to a function. For instance, you can add Alt+C to the keys for the copy function while retaining the standard Ctrl+Insert for compatibility.

Enable Menu Accelerators:

When this is checked, the standard menu accelerator keys such as Alt+F (to drop-down the File menu) take precedence over individual keystroke assignments. In general, this is the normal situation in Windows applications.

If you prefer to be able to use the normally preempted keystrokes for editing functions (for instance, if you want Alt+F to immediately save a file), you can clear this checkbox. You can still access the menus with the mouse, or by pressing and releasing Alt.

Keyboard Configuration File:

Load and **Save**:

When the pcAnywhere Editor starts, it reads a file named DEFAULT.KEY from its home folder to map keystrokes to functions. If you want to maintain more than one set of keystroke assignments, you can enter a new name in the Keyboard Configuration File text box and click Save. Then, to load that file at a later date, enter its name in this text box and click Load.

Changes you make are in effect only for the current session unless you use the Save button to save the assignments.

See Also

Customizing the pcAnywhere Editor

The Key Assignments command

The Editor Preferences command

The Document Preferences command

Key Assignments dialog box (Standard Browse)

Use this dialog box to locate an existing keyboard configuration file.

File Name: Type or select the desired filename. This box lists files with the extension selected in the List Files of Type box.

List Files of Type: Select the type of files to display in the File Name list box.

Keyboard Configuration Files (*.KEY): Lists all files in the current folder that were saved with a .KEY extension.

Drives: Select the drive on which the desired file is stored.

Folders: Select the folder in which the desired file is stored.

The Tool menu

This menu lets you access the script language quick reference dialog box. You can also access the Session Dial dialog box, used in writing scripts for unattended sessions.

Reference	Displays the Reference dialog box.

Reference command (Tool menu)

Function name:**batch_help**

Default key: **Alt+R**

This command displays the pcAnywhere Editor Reference dialog box, which provides an alphabetically ordered list of script language commands and a description, syntax summary, and example of each command.

You can keep the Reference dialog box open as you create or edit your script file. Use Alt+F6 to toggle back and forth between the Reference dialog box and the editing window.

Reference Dialog Box

Use this dialog box as a quick reference for the 150 commands in the pcAnywhere script language. This dialog box appears when you choose Reference from the Tools menu.

Commands

* Use and to scroll though this list.
* Or, press the first letter of any command to scroll quickly to that part of the list.

Description

As you scroll through the Commands list box, the Description box displays a short description of the command, its syntax and example of its use.

Close

Click this button to close the Reference dialog box.

Add

This inserts the currently highlighted statement from the Commands list into the text-editing screen at the current cursor position. This handy feature will help you avoid misspelling command names.

Assist

This button is enabled only when the Session Dial command is selected. Use the Assist button for help when you want to write a script to be used in an unattended session. The pcAnywhere Editor helps you create this file by taking the information you enter in the Assist dialog box and generating the basic commands required for the script. Use this dialog box to ensure that the proper parameters are set.

Dial Commands Dialog Box

Use when you want to write a script for commands to be used in an unattended session. The pcAnywhere Editor helps you create this file by taking the information you enter in the Assist dialog box and generating the basic commands required for the script. Use this dialog box to ensure that the proper parameters are set.

For specific information about this and other Script Language commands, see Chapter 4 of the Creating pcAnywhere Scripts manual.

Load Commands Dialog Box

Use when you want to write a script for commands to be used in an unattended session. The pcAnywhere Editor helps you create this file by taking the information you enter in the Assist dialog box and generating the basic commands required for the script. Use this dialog box to ensure that the proper parameters are set.

For specific information about this and other Script Language commands, see Chapter 4 of the Creating pcAnywhere Scripts manual.

Receive Commands Dialog Box

Use when you want to write a script for commands to be used in an unattended session. The pcAnywhere Editor helps you create this file by taking the information you enter in the Assist dialog box and generating the basic commands required for the script. Use this dialog box to ensure that the proper parameters are set.

For specific information about this and other Script Language commands, see Chapter 4 of the Creating pcAnywhere Scripts manual.

Send Commands Dialog Box

Use when you want to write a script for commands to be used in an unattended session. The pcAnywhere Editor helps you create this file by taking the information you enter in the Assist dialog box and generating the basic commands required for the script. Use this dialog box to ensure that the proper parameters are set.

For specific information about this and other Script Language commands, see Chapter 4 of the Creating pcAnywhere Scripts manual.

Wait Commands Dialog Box

Use when you want to write a script for commands to be used in an unattended session. The pcAnywhere Editor helps you create this file by taking the information you enter in the Assist dialog box and generating the basic commands required for the script. Use this dialog box to ensure that the proper parameters are set.

For specific information about this and other Script Language commands, see Chapter 4 of the Creating pcAnywhere Scripts manual.

Session Dial dialog box

Use this dialog box to write a script to be used in an unattended session. The pcAnywhere uses the information you enter here to generate correctly formatted Session commands. These commands contain the parameters used during the unattended session.

Once you have the session outlined, complete the script by specifying the action to occur after the connection has been made. These commands should be inserted into your script at the line COMMENT: ** insert SESSOPR here **. If you are using your script to launch an application on the host computer, be sure to use the correct SessOpr commands. The corresponding "regular" commands (such as Send File) in the script language are unavailable for unattended sessions and return an error when used.

Host to Call: Click the prompt button to select the name of the host you want to connect to during the session. The drop-down list box contains the same set of names as those in the Host PC Folder list box in the Call Host PC main window. For example, choosing Carol's Office PC from the list box generates the command:

```
SESSION DIAL "Carol's Office PC"
```

Cancel Host After Session: Check this check box to disable the host TSR when your session is complete. If this box is checked, the following command is generated:

```
SESSION HOST EXITMODE CANCEL
```

Call Immediately: Check this check box if you want the call to be made as soon as the script is executed. You can also specify a date and time at which the call is to be made. If this box is unchecked, then the Time to Call and Date to Call text boxes are enabled.

Time to Call: Enter the time you want pcAnywhere to dial the host. The format is military time (24-hour clock). If you specify a time, you must also specify a date in the Date to Call text box. For example, specifying a time of 10:00 P.M. on June 30, 1993 generates the command:

```
SESSION DIAL "Carol's Office PC" 22:00 93/06/30
```

Date to Call: Enter the date you want pcAnywhere to dial the host. The format is YY/MM/DD. If you specify a date, you must also specify a time in the Time to Call text box.

Retries: Enter the number of times you want pcAnywhere to redial the host. An entry of 3 generates the command:

```
SESSION RETRY 3
```

Time Between Retries: Enter the number of minutes that elapse between retries to dial the host. An entry of 10 generates the command:

```
SESSION DELAY 10
```

If Error Occurs: Click the prompt button to select the action to perform if an error should occur during the unattended session. You can select from:

END SCRIPT--ends the script and returns control to pcAnywhere.

CONTINUE SESSION--ignores the command which caused the error and advances to the next command.

END CURRENT SESSION--advances to the matching Session End command and continues processing from that point on in the script.

For example, selecting End Script generates the following command:

```
SESSION ONERROR END
```

File Transfer Overwrites: Click the prompt button to select the action to perform when the filename parameter in a SessOpr Host Send or SessOpr Remote Send command is the same as a filename at the destination. You can select from:

ALWAYS--causes the destination file to be overwritten.

NEVER--causes the file sent to be ignored.

OLDER--causes the file at the destination to be overwritten if the file being transferred is more recent (compares date/time stamp).

For example, selecting Older generates the command:

```
SESSION OVERWRITES OLDER
```

Error Time Out: Enter the time (in minutes) of the remote's inactivity timeout for unattended sessions. For example, entering 30 generates the command:

```
SESSION TIMEOUT 30
```

The Window menu

This menu lets you select, create, and manipulate the pcAnywhere Editor's editing windows.

New Window	Creates an additional window containing a copy of the file in the active window.
Cascade	Rearranges windows in a cascading pattern.
Tile	Rearranges windows in a non-overlapping pattern.
Arrange Icons	Aligns any minimized editing windows.
1 to *n*	Activates one of the editing windows, restoring it to its standard size if it is minimized.

See Also

Managing Multiple Windows
Window Control Functions

New Window command (Window menu)

Function name:**new_window**

Default key:**Alt+W,C**

Creates a new editing window and activates the new window.

If the same file is open in two or more editing windows, editing the file in any of the windows affects the contents of the file. Also, any file-related commands, such as changing the name using the Save As... command, affect all windows. However, the current cursor position, selection, and similar attributes are tracked separately for each window.

See Also

Managing Multiple Windows
Window Control Functions

Cascade command (Window menu)

Function name:**cascade**

Default key:**Alt+W,C**

Reorganizes all open editing windows in a cascading pattern. This usually makes it possible to see the titles of all the open windows, while making one entire window visible.

Any minimized windows remain icons.

See Also

Managing Multiple Windows

Window Control Functions

Tile command (Window menu)

Function name:**tile**

Default key:**Alt+W,T**

Reorganizes all open editing windows in a tiled (non-overlapping) pattern. This makes it possible to see at least a small part of all of the open windows.

Any minimized windows remain icons.

See Also

Managing Multiple Windows

Window Control Functions

Arrange Icons command (Window menu)

Function name:(none)

Default key:**Alt+W,A**

Aligns all minimized editing windows along the bottom of the pcAnywhere Editor application window.

See Also

Managing Multiple Windows

Window Control Functions

pcAnywhere Editor Keys

The pcAnywhere Editor lets you assign your own favorite keystrokes to any of its over 100 editing functions. See Customizing Editor Keystrokes for details.

For information on default keystroke assignments, review the topics below:

Cursor Movement

Selecting Text

Function Keys

Other Keys

See Also

The Key Assignments command
Using Editor Macros

Scripts dialog box

From the Scripts dialog box, you can create a new script, or select a script you have already created and edit, compile or run it.

The Scripts list box contains a list of all the currently defined scripts and a short description of each. Directly beneath the list box is the Script Path text box. If your script files are located in a folder other than the one displayed, you can change directories using the Select… button.

After highlighting a file in the list box, simply click the Edit, Compile or Run button to perform that function on the selected script.

Select Folder dialog box

Use this dialog box to specify the drive and folder where your script files are located.

Folders: Scroll through the list box to locate the folder in which your script files are located.

Drives: Click the prompt button to display a list box containing all available drives.

All script files in the current script path appear here. Select a script and click Run, Compile or Edit.

Displays the currently selected path. If your script files are in a different folder, click Select and select a new drive and folder.

Click here to display the Select folder dialog box and change your Script Path.

Select a script from the Scripts list box and click here to execute the selected script.

Select a script from the Scripts list box and click here to compile the selected script.

Select a script from the Scripts list box and click here to edit the selected script.

Click here to launch the pcAnywhere Editor and create a new script.

Click here to close the dialog box and return to the pcAnywhere main window.

New command (File menu)

Use the New command to create a new document file.

Open command (File menu)

Use the Open command to open an existing document file.

Close command (File menu)

Use the Close command to close active document window.

Save command (File menu)

Use the Save command to save the file in the active window.

Save As command (File menu)

Use the Save As command to save the file in active window under a different name.

Save All command (File menu)

Use the Save All command to save all currently open files.

Insert command (File menu)

Use the Insert command to insert the contents of a file at the cursor position.

Revert command (File menu)

Use the Revert command to undo all changes made since the last save.

Write Block command (File menu)

Use the Write Block command to write the selected text to a file you specify.

Page Setup command (File menu)

Use the Page Setup command to set headers, footers, margins, etc.

Print command (File menu)

Use the Print command to send the contents of the file in the active document window to the printer.

Compare command (File menu)

Use the Compare command to compare the contents of two documents.

Exit command (File menu)

Use the Exit command to close the pcAnywhere Editor windows and exit.

(list of files) command (File menu)

Select any number from 1 to 4 to open or activate the recently used file asociated with that number.

Undo command (Edit menu)

Use the Undo command to undo the most recent text-editing change.

Cut command (Edit menu)

Use the Cut command to copy the selected text to the clipboard and then delete the text from the document.

Copy command (Edit menu)

Use the Copy command to copy the selected text to the clipboard.

Paste command (Edit menu)

Use the Paste command to paste the contents of the clipboard to the current cursor position.

Delete command (Edit menu)

Use the Delect command to delete the current selection.

Select All command (Edit menu)

Use the Select All command to select all text in the active file.

Time/Date command (Edit menu)

Use the Time/Date command to insert the current date and time at the current cursor position.

Wrap Paragraph command (Edit menu)

Use the Wrap Paragraph command to reformat the paragraph at the cursor, wrapping at the current margin settings.

Word Wrap command (Edit menu)

Use the Word Wrap command to toggle automatic paragraph wrapping on or off.

Record Macro command (Edit menu)

(toggles to Stop Recording Macro) Use the Record Macro command to record subsequent keystrokes for automatic recall.

Playback Macro command (Edit menu)

Use the Playback Macro command to replay the keystrokes saved during the most recent Record Macro command.

Find command (Search menu)

Use the Find command to search for text or a regular expression.

Find Again command (Search menu)

Use the Find Again command to continue the search started with the Find command.

Replace command (Search menu)

Use the Replace command to search for and replace text or regular expression.

Find Files Containing command (Search menu)

Use the Find Files Containing command to generate a list of files which contain specified text or regular expression.

List Found Files command (Search menu)

Use the List Found Files command to see a list of files found; choose one to edit.

Goto Line command (Search menu)

Use the Goto Line command to move the cursor to a line specified by its line number.

Document Preferences command (Options menu)

Use the Document Preferences command to set the margins, tabs, indent, etc. for current document only or the default for all documents.

Editor Preferences command (Options menu)

Use the Editor Preferences command to set the screen font, cursor, backup, and autosave options.

Key Assignments command (Options menu)

Use the Key Assignments command to customize keystrokes for fast access to editor functions.

Reference command (Tool menu)

Use the Reference command to display the script language Reference dialog box. The Reference dialog box contains the name of each of the script langauge commands, a description, syntax and an example of its use.

New Window command (Window menu)

Use the New Window command to create an additional window on the document in the active window.

Cascade command (Window menu)

Use the Cascase command to rearrange the document windows in a cascading pattern.

Tile command (Window menu)

Use the Tile command to rearrange the document windows in a non-overlapping pattern.

Arrange Icons command (Window menu)

Use the Arrange Icons command to align any minimized document windows.

1 to n command (Window menu)

Select a number from 1 to n to activates one of the document windows, restoring it to its standard size if it is an icon.