INTRODUCTION

Winsock RCP/RSH/REXEC for Win32 includes versions of the standard rcp, rsh, and rexec client utilities for Windows 95/98/ME and Windows NT/2000/XP. These utilities originated on the Unix operating system.

Winsock RCP allows you to copy files to and from a remote host over TCP/IP. The remote host must be running a *remote shell daemon* (*rshd*) that supports rcp copies. This can be a Unix system or it can be another Windows system running Denicomp Systems Winsock RSHD, Winsock RSHD/95, or Winsock RSHD/NT. Windows does not include a remote shell daemon.

Winsock RCP differs from other file transfer utilities, such as FTP, in two ways. First, it is non-interactive; filenames and options are specified entirely on the command line, much like the COPY command or Unix cp command. Second, it does not require you to specify a password for the remote host. Security is enforced through *host equivalence*, which is explained later.

Winsock RSH allows you to execute a command on a remote host over TCP/IP. The remote host must be running a *remote shell daemon* (*rshd*). You can view any standard output and standard error of the command executed or capture it to a file. Generally, rsh should not be used to execute interactive commands; use *telnet* or *rlogin* if this is what you require.

Winsock REXEC is very similar to Winsock RSH. It allows you to execute a command on a remote host over TCP/IP. The remote host must be running a *remote exec daemon* (*rexecd*). The difference between Winsock RSH and Winsock REXEC is that RSH does not require you to specify a password; security is enforced through *host equivalence*, which is explained later. REXEC does not require host equivalence, but does require you to specify a valid password for the remote host.

REQUIREMENTS

Winsock RCP/RSH/REXEC for Win32 requires a PC running Windows NT, Windows 2000, Windows XP, Windows 95, Windows 98, Windows ME, or other Windows operating system that supports 32-bit Windows Sockets and the Microsoft Win32 API. You must also have a connection via TCP/IP to a host running the *rshd* (Remote Shell Daemon) service and/or the *rexecd* (Remote Execution Daemon).

INSTALLATION

To install Winsock RCP/RSH/REXEC from diskette, use the following procedure:

1. Insert the Winsock RCP/RSH/REXEC diskette into your diskette drive.

2. Click the ***Start*** button and select ***Run*** (or use ***File/Run*** if using Windows NT 3.51).

3. Type the drive containing the Winsock RCP/RSH/REXEC diskette followed by **SETUP** and press Enter.

For example, if the diskette is in the A: drive, type **A:SETUP** and press Enter. Note that there is no space between the colon (:) and word SETUP.

If you downloaded a .EXE file containing the registered version of Winsock RCP/RSH/REXEC for Win32, simply execute the .EXE file by using the Windows Explorer and double-clicking on that file.

This will create the Winsock RCP/RSH/REXEC program group.. It will contain selections for Visual RCP, Visual RSH, and Visual REXEC along with options to view and/or print the manual in ASCII format.

REMOVING WINSOCK RCP/RSH/REXEC

To uninstall Winsock RCP/RSH/REXEC, use the **Add/Remove Programs** applet in the Control Panel.

To manually remove Winsock RCP/RSH/REXEC, remove all of the files in the directory you selected when you installed the software and also the file RCPRSH32.CPL in the Windows System directory (\WINDOWS\SYSTEM for Windows 95/98/ME, \WINNT\SYSTEM32 for Windows NT/2000/XP).

SECURITY

First, a warning. The use of rsh and rcp on your network can potentially leave your systems open to unauthorized access. These utilities were originally designed to be used within a network of Unix systems where users were generally trusted and convenience was more desirable over tighter security.

While the use of the rcp, rsh, and rexec commands will not compromise the security of your PC, the use of the remote shell daemon on your systems can leave those systems vulnerable. If your systems are not externally accessible via the Internet, your only concern would be the users on your internal network. If your systems are connected to the Internet with public IP addresses, the use of products like firewalls, TCP wrappers, etc. can help to prevent unauthorized access.

If the remote host is a Unix system, you must establish ***host equivalence*** before you can use Winsock RSH or Winsock RCP to access that host. The rules for establishing host equivalence is determined by the remote shell daemon (rshd) on the remote host and not by Denicomp Systems. We cannot alter these requirements.

Host equivalence is not required to use Winsock REXEC, since it requires you to specify a password. Only rsh and rcp require host equivalence.

The Unix remote shell daemon uses the following steps to determine whether or not you should be granted access:

- It verifies that your user exists in the */etc/passwd* file on the remote host. By default, the user login used when logging in to Windows is used , but this can be overridden either on the rsh or rcp command line or using the **R-Commands** Control Panel applet. This is explained later. If it is not a valid user login, the host denies access.

  Some Unix systems will deny access if the user's password has expired or if the user has no password specified.

- It tries to change to the user's home directory on the remote host. If it cannot (because the directory is missing or permissions do not allow it), access is denied.

- It looks up the host name of your PC based on its IP address. The host name and IP address of your PC must be in the */etc/hosts* file on the remote host or must be able to be resolved by the remote host using DNS (Domain Name Service). If it cannot find a name associated with your PC's IP address, access is denied.

- It looks in the file */etc/hosts.equiv* file on the remote host to see if your PC's host name is listed.

- If there is a line in the */etc/hosts.equiv* file that contains only a plus sign (+), access is granted. This allows **all** hosts and users access, as long as the previously listed criteria is met. Use this method with caution, especially if your host system is on the Internet.

- If the name is listed on a line by itself, access is granted.

- If the host name is listed in */etc/hosts.equiv* along with a user name in the format:

hostname username

access will only be granted from that host when using that user name.

If the user's name list listed in */etc/hosts.equiv* prefixed by a plus sign (+), access will be granted to anyone using that user name regardless of the host. For example:

+ johns

would give the user "johns" access from any system without a password.

- If access is not granted through the /*etc/hosts.equiv* file, it looks in your home directory on the remote host for a file named *.rhosts*. The *.rhosts* file **must** be owned by either you or root, and only the owner should have read and write access. That is, the permissions on the file **must** be **0600**. It then searches that file using the same steps listed for searching */etc/hosts.equiv*.
- If access is not granted through either */etc/hosts.equiv* or *$HOME/.rhosts*, access is denied and you will receive an error from the remote host.

If you use DHCP (Dynamic Host Configuration Protocol) and your PC is assigned a different IP address each time you boot or your ISP (Internet Service Provider) gives you a different IP address each time you connect, each possible IP address along with a host name must be listed in */etc/hosts* or be resolvable through DNS. Additionally, each of those host names must be listed in either */etc/hosts.equiv* or *$HOME/.rhosts.*

This may be feasible when using DHCP if the range of IP addresses is not large, but may not be possible when using an ISP. If it is not feasible to list each possible host name and IP address, your only choices are to list the user names prefixed by a plus sign (+) or place a plus sign on a line by itself at the beginning of the file. However, keep in mind that this will make your system extremely vulnerable to unauthorized access. Another option would be to obtain fixed IP addresses from your ISP so your PC gets the same IP address each time; this is sometimes available at an additional cost.

For additional information, type ***man rshd*** at a Unix prompt to read the documentation on the Unix rshd.

RSH/REXEC AND FIREWALLS

There are some unique characteristics about the rsh and rexec protocols that can cause problems with firewalls.

By default, the standard rsh and rexec clients use **two** connections to the server. The first connection is established from the client (where the rsh or rexec command is issued) to the server. If the rsh command is used, this will connect on port 514 and if rexec is used, this will connect on port 512. It will send the user information and remote command to the server on this connection. Later, it will use this connection to exchange stdout and stdin data.

The commands then look for an unused local port number to be used for stderr data. When one is found, this is transmitted to the server and then the rsh/rexec client will **listen for a return connection from the server on that port**. This is important to understand – this second connection will be in the **opposite** direction from the first connection. The server will connect back to the client on this port. If a firewall blocks the server from connecting back to the client for this second connection, you will receive an error and the remote command will not execute.

When using rsh, this second connection will always be on a port number between 513 and 1023. It will start with port 1023 and work downwards toward 513 until it finds an unused local port number. When using rexec, the second connection will use a random port number above 1024.

If security concerns require that you have ports blocked from the server back to the client, the only option is to try to use the **–c** option in the rsh or rexec commands. The **–c** option tells rsh or rexec to combine stdout and stderr data onto the first connection and the second connection will not be used. This usually solves problems with firewalls. Note however, you will not be able to separate data sent to stdout and stderr on the client, so they cannot be redirected separately.

CONFIGURING WINSOCK RCP/RSH/REXEC

Winsock RCP/RSH/REXEC does not *require* any configuration after it is installed. However, you may want to configure it to make its use more convenient for you.

Configuration is done through the **R-Commands** applet in the Control Panel. Almost all options found there can be controlled using command line parameters. If there are certain options that you require consistently, they can generally be enabled using the Control Panel.

When using the **R-Commands** applet to configure Winsock RCP/RSH/REXEC, you will see six (6) tabs across the top of the window. They are:

RSH – All Users	**RSH – Current User**
REXEC – All Users	**REXEC – Current User**
RCP – All Users	**RCP – Current User**

The options on pages labeled **All Users**, if set, apply to all users logging in on this system. The options on pages labeled **Current User**, if set, apply only to the currently logged in user on this system. This distinction is only significant if multiple users log in on this system (using the Windows logon).

The same options are available on the **All Users** and **Current Users** pages. Winsock RCP/RSH/REXEC first checks the options set for the **Current User**, then looks to the options set for **All Users** if the option is not set for the current user.

Below is a brief description of each of the options available. Only brief descriptions are given; see the sections on the command line options available for each utility for more details. For items that only set a certain command line option, that option is given here; read more about the option in the Command Line Options description later.

WINSOCK RSH CONFIGURATION OPTIONS

Remote User	**USE THIS USER LOGIN AT THE REMOTE HOST INSTEAD OF THE NAME USED WHEN LOGGING IN TO WINDOWS. THIS CAN BE OVERRIDDEN ON THE COMMAND LINE WITH THE –L OPTION.**
Startup Window	The appearance of the window when using wrsh. The options are Normal, Minimized, and Hidden. This has no effect on the rsh command; it only affects wrsh.
ASCII Stdout/Stdin/Stderr	Sets the –a option.
Binary Stdout/Stdin/Stderr	Sets the –b option.
ASCII Stdout	Sets the –o option.
ASCII Stdin	Sets the –d option.
ASCII Stderr	Sets the –e option.
Wait at End	Sets the –w option.
Disable Command Line Macros	Disables the use of command line macros, which begin with the at-sign (@). Command line macros allow you to store remote commands in a local file. You can disable them if you have remote commands that happen to begin with the at-sign (@).
Use Blocking Sockets	This is an advanced option that should only be enabled at the direction of Denicomp Systems. If you check this option, the Timeout options are effectively disabled.
Always Send Login User (like Unix)	The rsh protocol sends the host system two user names to be validated, a local user and a remote user. Normally, Winsock RSH sends the same user name for both the local and remote user. The Unix rsh always sends the current login user name as the local user, regardless of whether it is overridden using the -l option. Checking this option makes Winsock RSH act like the Unix rsh in this regard. When checked, it will always send the current Windows login name as the local user, regardless of whether the -l option is used or if a user is specified in the Remote User field above. The Windows login name must be a valid user on the host system.
Host Lookup Timeout	The number of seconds to wait for a response from a DNS server to resolve the host name supplied on the command line. The default is 30 seconds. If no response is received in that amount of time, an error is reported and the command is not executed.
Connect Timeout	The number of seconds to wait for a connection to be established with the remote host. The default is 60 seconds. If the connection to the host is not established in that amount of time (for example, the host is down), an error is reported and the command is not executed.
Retries	The number of times to retry establishing the connection to the host if there is a failure. The default is to try only once. Use in combination with the Connect Timeout option. The default Connect Timeout is 1 minute, so setting this will require that many minutes before it gives up on trying to connect to the host. Reduce the Connect Timeout when using this.
NT/SP4 Connect Timeout Workaround	Enables a workaround that may help some users who have experienced connection timeout problems under Windows NT with Service Pack 4 (SP4) or later installed. Some users have reported that starting with SP4, rsh/rcp/rexec commands issued within 30 seconds of each other results in a connection timeout error. This occurs with the native Windows NT rsh/rexec/rcp commands also, so the problem lies in the

	NT TCP/IP stack.
Receive Timeout	The number of seconds to wait for data to be received from the remote host before reporting a timeout error. By default, there is no receive timeout. If you enter a receive timeout, data must be received within that timeframe from the remote command or an error will be reported and the connection will be closed. If you execute commands that can take a long period of time to complete without writing any data to stdout or stderr, you should not enter a value here. RSH will not be able to tell whether no data is received because the connection was lost or because the remote command is taking a long time to complete.
Send Timeout	The number of seconds to wait for data to be sent to the remote host before reporting a timeout error. The default is 60 seconds.
Port	The TCP/IP port to use for the RSH connection. The standard RSH port is 514. If you enter a different value here, the remote shell daemon (rshd) on the remote host must also be changed to listen on the port you specify.
Trace Output Level	If you enter a Trace Output File name, this determines the amount of output RSH will generate in the file. A value of zero (0) generates no output. A value of one (1) generates the least amount of output, two (2) generates more, etc.
Trace Output File	Enter a filename here and a Trace Output Level to cause RSH to generate trace information that may help you or Denicomp Systems support to pinpoint a problem.
Use WIN.INI Instead of Registry	If you check this option, RSH will look in the [RSH] section of the WIN.INI file in the Windows directory for options that would normally be stored in the Windows Registry. You should generally not use this option. It is included only for backward compatibility with older versions of RSH, but if you do use it, you must use a text editor to edit WIN.INI and specify the entries in the [RSH] section. If you require this, Denicomp Systems can provide a list of the entries you can include in WIN.INI. (They are the same as the registry entries created by this Control Panel applet.)

WINSOCK REXEC CONFIGURATION OPTIONS

The Winsock REXEC Configuration Options are basically the same as those available for Winsock RSH, except for the following additional option:

Password	Enter the password for the user specified in the Remote User field. This password will be used at the remote host for that user and REXEC will not ask you to enter it each time.

Additionally, the Port option defaults to the value 512, which is the standard rexec port. If you check the Use WIN.INI Instead of Registry option, REXEC will look in the [REXEC] section of WIN.INI.

WINSOCK RCP CONFIGURATION OPTIONS

Remote User	Use this user login at the remote host instead of the name used when logging in to Windows. This can be overridden on the command line with the User@ option in the source or destination filename.
Password	If using RCP through REXEC (RCP/X with the –x option), use this password for the remote user when logging in to the remote host. If not using –x, this field has no meaning.
Startup Window	The appearance of the window when using wrcp. The options are Normal, Minimized, and Hidden. This has no effect on the rcp command; it only affects wrcp.
ASCII End-of-Line Conversion	Sets the –a option.
Recursively Copy	Sets the –r option.
Verbose – Display Filenames	Sets the –v option.
Preserve Filename Case	Sets the –c option.
Convert Destination Filenames to Lowercase	Sets the –l option.
Wait at End	Sets the –w option.
Preserve Modification Times	Sets the –p option.
Wildcards Match Hidden Files	Sets the –f option.
Wildcards do not Match Directories	Sets the –d option.
Send RCP through REXEC (RCP/X)	Sets the –x option.
Use NT's RCP Syntax	Tells RCP to use a command line syntax similar to Windows NT/2000/XP native RCP implementation for compatibility purposes. Instead of using a host/user format of User@Host:filename, the Windows native RCP uses the format Host.User:filename . This option tell RCP to detect and use the Windows format. It also enables the ASCII conversion option by default (-a) as the native rcp does.
Use Blocking Sockets	This is an advanced option that should only be enabled at the direction of Denicomp Systems. If you check this option, the Timeout options are effectively disabled.
Always Send Login User (like Unix)	The rcp protocol sends two user names to be validated, a local user and a remote user. Normally, Winsock RCP sends the same user name for both the local and remote user. The Unix rcp always sends the current login user as the local user, regardless of whether it is overridden using the user@filename construct. Checking this option makes Winsock RCP act like the Unix rcp in this regard. When checked, it will always send the current Windows login name as the local user, regardless of whether the "user@" is used or if a user is specified in the Remote User field above. The Windows login name must be a valid user on the host system.
Host Lookup Timeout	The number of seconds to wait for a response from a DNS server to resolve the host name supplied on the command line. The default is 30 seconds. If no response is received in that amount of time, an error is reported and the command is not executed.
Connect Timeout	The number of seconds to wait for a connection to be established with the remote host. The default is 60 seconds. If the connection to the host is not established in that amount of time (for example, the host is down), an error is reported and the command is not executed.
Retries	The number of times to retry establishing the connection to the host if there is a failure. The default is to try only once. Use in combination with the Connect Timeout option. The default Connect Timeout is 1

	minute, so setting this will require that many minutes before it gives up on trying to connect to the host. Reduce the Connect Timeout when using this.
NT/SP4 Connect Timeout Workaround	Enables a workaround that may help some users who have experienced connection timeout problems under Windows NT with Service Pack 4 (SP4) or later installed. Some users have reported that starting with SP4, rsh/rcp/rexec commands issued within 30 seconds of each other results in a connection timeout error. This occurs with the native Windows NT rsh/rexec/rcp commands also, so the problem lies in the NT TCP/IP stack.
Receive Timeout	The number of seconds to wait for data to be received from the remote host before reporting a timeout error. By default, there is no receive timeout. If you enter a receive timeout, data must be received within that timeframe from the remote host or an error will be reported and the connection will be closed.
Send Timeout	The number of seconds to wait for data to be sent to the remote host before reporting a timeout error. The default is 60 seconds.
Port	The TCP/IP port to use for the RCP connection. The standard RCP port is 514 when using the rsh protocol and 512 when using the rexec protocol. If you enter a different value here, the remote shell daemon (rshd) on the remote host must also be changed to listen on the port you specify.
Trace Output Level	If you enter a Trace Output File name, this determines the amount of output RCP will generate in the file. A value of zero (0) generates no output. A value of one (1) generates the least amount of output, two (2) generates more, etc.
Trace Output File	Enter a filename here and a Trace Output Level to cause RCP to generate trace information that may help you or Denicomp Systems support to pinpoint a problem.
Use WIN.INI Instead of Registry	If you check this option, RCP will look in the [RCP] section of the WIN.INI file in the Windows directory for options that would normally be stored in the Windows Registry. You should generally not use this option. It is included only for backward compatibility with older versions of RCP, but if you do use it, you must use a text editor to edit WIN.INI and specify the entries in the [RCP] section. If you require this, Denicomp Systems can provide a list of the entries you can include in WIN.INI. (They are the same as the registry entries created by this Control Panel applet.)

CONFIGURATION NOTES

For your convenience, if you need to specify a remote user override in the Control Panel, you can specify it in the **Remote User** field in the **RSH – All Users** or **RSH – Current User** page and leave that field blank in the **REXEC** and **RCP** pages. The rexec and rcp commands will look at the remote user specified in the **RSH** page if it is not specified in the command’s own page. This allows you to specify the remote user only one time in the **RSH** page instead of specifying it three times for each command.

VISUAL R-COMMANDS

Winsock RCP/RSH/REXEC for Win32 includes both command-line versions of the rcp, rsh, and rexec utilities and "visual" versions. The visual versions simply provide a fill-in-the-blanks interface for each utility. These can be found on your ***Start*** menu under **Winsock RCP/RSH/REXEC**. At installation time, you can choose to not install these if you are only interested in the command-line versions.

The following provides a brief description of the options available in the visual version of each utility. For a deeper understanding, you should review the documentation on the command-line versions.

Visual RSH allows you to execute a non-interactive command on a remote host and view the output of the program or store the output in a file. In Visual RSH, you must provide the following information:

Host	Enter the host name or IP address of the remote host where the command will be executed.
User	Enter the user login name to use at the remote host. You can leave this blank; if you leave it blank, it will use the login name used when logging into Windows unless you have specified a remote user override using the R-Commands Control Panel applet.
Command	Enter the command to execute on the remote host.
Output to Screen/File	If you want to view the output of the command you execute on the screen, check the Screen option; uncheck it if you do not. If you want to store the output of the command in a file on your system (not the remote host), check the File option. You may check both options to both view the output and store it in a file.
Output to File	If you checked the above option to store the output of the command in a file, enter the filename here.

Visual REXEC allows you to execute a non-interactive command on a remote host and view the output of the program or store the output in a file. In Visual REXEC, you must provide the following information:

Host	Enter the host name or IP address of the remote host where the command will be executed.
User	Enter the user login name to use at the remote host. You can leave this blank; if you leave it blank, it will use the login name used when logging into Windows unless you have specified a remote user override using the R-Commands Control Panel applet.
Password	Enter the password for the user specified. You can leave this blank if you specified the password in the R-Commands Control Panel applet.
Command	Enter the command to execute on the remote host.
Output to Screen/File	If you want to view the output of the command you execute on the screen, check the Screen option; uncheck it if you do not. If you want to store the output of the command in a file on your system (not the remote host), check the File option. You may check both options to both view the output and store it in a file.
Output to File	If you checked the above option to store the output of the command in a file, enter the filename here.

Visual RCP allows you to copy files to or from a remote host using the RCP protocol. In Visual RCP, you must provide the following information:

Copy From	Enter the filename(s) to copy. If you are copying files from your system to a remote host, you can simply enter the name(s) of the files. You can use wildcards (* and ?) and you can specify multiple files by separating the names with spaces. You can also use the **Select** button to select the files to copy. If you are copying files from the remote host to your system, enter the name(s) of the files and prefix each with the host name and a colon (e.g. **remhost:/dir/file**). You can use wildcards and specify multiple files by separating the names with spaces (but you must specify the host for each name).
Copy To	Enter the destination of the files. If you are copying files from a remote host to your system, you can simply specify the name of a file or directory. If you are copying files from your system to a remote host, specify the destination file or directory and prefix it with the host name and a colon (e.g. **remhost:/dir/file**). If you are copying a single file, the destination can be either a directory or the name of a file. If you are copying multiple files (using wildcards or two or more filenames), the destination must be a directory.
ASCII End-of-Line Conversion?	Check this option if you want the end-of-line characters converted to those required on the destination system (NL only for Unix, CR/NL for Windows).
Recursive Copy?	Check this option if you specified a directory in the **Copy From** field and you want to copy all files in that directory and all subdirectories under that directory. Without this option, only the files in that directory are copied.
Preserve File Access/Modification Times?	Check this option if you want the files created on the destination system to have the same access and modification times as the source files in their directory entries. Otherwise, they will be created with the current date and time.
Replace Spaces in Filenames with Underscores?	Check this option if you want RCP to replace any spaces in filenames created on the destination system with the underscore character (_). This is useful if the destination system has problems with spaces in filenames and the file(s) copied have spaces in their names.

WINSOCK RCP

Winsock RCP ("Remote Copy") copies files between a PC and a remote host or between two remote hosts. It is similar to the Unix utility of the same name.

There are two versions of Winsock RCP included: a "console" version that is well-suited for use from the Windows Command Prompt (named **rcp**) or batch files and a windowing version that displays output in a window, allowing you to use scroll bars to review the output, cut/paste it, or save it to a file. The windowing version is named **wrcp**.

The remote must be a system running *rshd* (remote shell daemon). This can be a Unix system, a Windows system, or another operating system with a Unix compatible RSHD. Note that Windows does not come with a remote shell daemon. You can use Denicomp Systems' Winsock RSHD, Winsock RSHD/95, or Winsock RSHD/NT.

Please note that if you are using Windows NT, 2000, or XP, it includes its own **rcp** command. Since the **rcp** included in this package has the same name, you must be sure to put the directory that contains Winsock RCP at the beginning of your PATH environment variable (or rename the Windows **rcp** command). The primary differences between Winsock RCP and the rcp supplied with Windows are:

- The native Windows' rcp defaults to copying with end-of-line conversion of files. You must use the –b (binary) option if you do not want any translation of the data in the file. Winsock RCP defaults to binary copies with no conversion and will only perform the end-of-line conversion with the –a (ASCII) option.

- The native Windows rcp does not use the user@host syntax when specifying the user along with the host and filename. It uses a different format; see the Windows documentation for more details.

SYNTAX

Console Version:

```
rcp [-abcdflmprsvwx] [-n password] source ... destination
```

Windowing Version:

```
wrcp [-abcdfhlmprsvwx] [-n password] source ... destination
```

Source and Destination Syntax:

```
[[User@][Host:]]{File|[Dir]}
```

HOST AND FILENAME SPECIFICATION

User@

(optional) Specifies the user name to be used at the remote host. If this prefixes the ***Host:*** parameter, this user name overrides the name of the user currently logged in.

Host:

Specifies the host name of the remote host. This is not required if the file or directory referenced is on the local system. This host must be a system running the remote shell daemon (*rshd)* process.

File

Specifies the filename of the source or destination file. You may use wildcard characters to copy multiple source files. You may also specify multiple source files individually by separating the names with spaces. If the filename includes embedded spaces, you must enclose it in double quotes.

Dir

Specifies the name of the source or destination directory. If you specified multiple source filenames or used wildcard characters in the source filename, the destination **must** be a directory.

HOST NAMES

The **Host:** parameter is required for either the source file/directory or the destination file/directory. Copying between two remote hosts is permitted by specifying a Host: for both the source and destination.

-

The **Host** may be either a name or an IP address. If you use a name, it must be resolvable by your system; that is, it must either appear in the HOSTS file or be resolvable through a DNS (Domain Name Service) server.

If a remote host name is not specified for either the source or the destination, you will receive an error. Use the COPY command instead. Do not use **Host:** parameter when referencing local files.

USER NAMES

The user name (or login name) determines the file access privileges permitted at the remote host. This name also determines the ownership and access modes of the destination file or files.

If the **Host:** is prefixed by the **User@** parameter, that name is used as the user name at the remote host. If the **User@** parameter is not used, the local user name is used at the remote host.

The local user name is normally the name you used when logging in to Windows. For example, if you logged in to Windows as the user "joed", Winsock RCP will use "joed" as the user name at the remote host. Winsock RCP will always convert this name to all lowercase characters.

You can override the Windows login user name when using Winsock RCP by specifying an alternate user in the **R-Commands** applet in the Control Panel. Specify the alternate user either the **RCP – All Users** tab or the **RCP – Current User** tab. The value entered in the **Current User** tab will only be in effect when you log into Windows; it allows you to specify other user names for other users if multiple people use your system.

FILENAMES

If a full directory path is not specified for a remote host, the path begins at the user's home directory. That is, if the file/directory name specified after the **Host:** parameter does not begin with a slash (/), it is assumed to reference a file/directory in the user's home directory.

For example, the file "joe@remhost:file.txt" refers to the file "file.txt" in the home directory of the user "joe" on the host "remhost".

Filenames may contain either slashes (/) or backslashes (\) as directory separators, for either the host file/directory or file/directories on the PC. They will be converted to the appropriate separator.

You can copy multiple files by using wildcard characters, such as * or ?. The wildcard characters must be valid for the appropriate source system. For example, if you are copying files from a Windows system, the only valid wildcard characters are * and ?. However, if you are copying files from a Unix system, you can use the full range of wildcard characters available on Unix in the source specification, since they will be interpreted at the host.

You can also copy multiple source files by separating them with spaces. If the source files reside on the remote host, you must specify the Host: (and optionally the User@) parameter for each file.

If you copy multiple source files with wildcard characters or by specifying individual filenames, the destination **must** be a directory.

Note that a colon (:) terminates the host name. This causes a problem when filenames on the PC require a drive letter (e.g. A:). If a file name specification begins with one character between A and Z and is followed by a colon (:), Winsock RCP will interpret this as a drive letter instead of a host name. This means that Winsock RCP cannot handle one character host names.

-

The destination cannot contain only a drive specification (e.g. A:). It must also include a filename or a directory name. If the destination is the current directory on the drive, use "." (e.g. A:.); if the destination is the root directory on the drive, use “\” (e.g. A:\).

COMMAND LINE OPTIONS

-a	ASCII conversion. For file(s) transferred to the remote host, all sequences of CR/NL (Carriage Return/New Line, ASCII 13/10) will be converted to NL (ASCII 10). This is the standard Unix text file format. For file(s) transferred from the remote host to the PC, a CR (ASCII 13) will be added before every NL (ASCII 10) if that NL is not already preceded by a CR. This is the standard MS-DOS text file format. Without the -a option, files are transferred with no translation (binary).
-b	Binary transfer. File(s) are transferred with no modifications. This is the default, so this option is not necessary if you want a binary transfer. It is provided for command line compatibility with other RCP commands from other vendors.
-c	Preserve the case of filenames copied using the recursive option (-r) or wildcards. By default, all filenames are converted to lowercase characters before being sent to the remote host. This is normally useful, especially when copying to case-sensitive systems, such as Unix. Since Windows can store filenames in mixed case (even though the file system is not case sensitive), you can use this option to preserve the case of the names as they appear in the Windows directory. This **only** affects recursive copies and wildcard copies. If you copy individual files by specifying their names on the command line, the case that you use on the command line will be used at the remote host. For example: rcp xyz unix:/tmp creates the file "/tmp/xyz" on the "unix" host. The command: rcp XYZ unix:/tmp creates the file "/tmp/XYZ" on the "unix" host. Note that under Windows, "xyz" and "XYZ" represent the SAME file. However, under Unix, these are two different filenames.
-d	Ignore directories in wildcards. When using wildcards to copy multiple files, normally RCP will attempt to copy any directories that match the wildcard pattern. If you are not using the recursive copy option (-r), you will receive errors on the directory names matched. This option excludes the directories when matched and you will not receive the errors.
-f	Copy hidden files. Normally, wildcard patterns and recursive copies to not include files with the "hidden" attribute. If you specify this option, wildcard copies and recursive copies will copy hidden files.
-h	Run Hidden. Like -m, but the window will be completely hidden. Use with caution, since hidden windows cannot be accessed using the Task Manager, so you cannot use it to manually stop a transfer. This is only available in the windowing version (wrcp). Also, it **must** appear before any option that requires an additional argument (such as -n).
-l	Convert filenames to lowercase on the remote host. When this option is used, filenames will be converted to lowercase and created on the remote host in lower case, regardless of how they are specified on the command line. Without this option, the case used on the command line is used at the remote host (unless wildcards are used).
-m	Windowing version. Run Minimized. Normally, Winsock RCP will display a window showing progress and any possible error messages. With -m, Winsock RCP will only display a minimized icon while running. This is useful for software developers running Winsock RCP transparently

	from within their software. This is only available in the windowing version (wrcp). Also, it **must** appear before any option that requires an additional argument (such as -n).
-m perms	Console version. Allows you to specify file permissions in octal that will override any file permissions rcp sends to the server. Normally rcp sends file permissions to the server based on a translation of Windows permissions to Unix permissions (read only, executable, etc.). This option allows you to override that translation. For example, "-m 644" would create the files on the server (assuming the server is Unix) with permissions of 644 (-rw-r--r--), regardless of Windows permissions.
-n password	Use the *rexec* protocol for the data transfer and use **password** at the remote host when logging in. This is similar to the **–x** option, but allows you to specify the password on the command line; the **–x** option will ask for it interactively. See the section later on “RCP through REXEC” for more details.
-p	Preserve access and modification times. Normally, when you copy a file to the local system, the files are created with access and modification times of the current date and time. This option will preserve the access and modification time of the file as it was on the remote host.
-r	Recursively copies, for directories only, each file and subdirectory in the source directory into the destination directory.
-s	Convert spaces in filenames to underscores (_). If this option is specified, any spaces found in filenames copied will be changed to an underscore. This is useful if the destination system cannot handle filenames with spaces or it is inconvenient to do so. The spaces are converted to underscores in the filename on the destination system only. The spaces in the original filename are not affected.
-v	Verbose. In the Console version (rcp), this will display the name of each file as it is copied. Normally, this does not occur. In the windowing version (wrcp), the names of the files display in the window automatically, so this option is not necessary, but specifying it will not cause an error.
-w	Wait at the end of the command. In the Console version (rcp), it will wait for you to press the Return key after the file(s) are copied so you can view the list of files copied and any error messages from the remote host In the windowing version (wrcp), it will wait for you to close the RCP window (using File/Exit from the menu, for example). If you are copying multiple files and the list scrolls the window, you can use the scrollbars to review the scrolled information. You can also cut/paste information into other windows or save the information in the window to a file
-x	Use the *rexec* protocol for the data transfer. This will cause RCP to ask you to type in a password to be used to log into the remote host. When the default *rsh* protocol is used, no password is required. See the section later on “RCP through REXEC” for more details.

EXIT CODES

RCP will return an exit code value of zero (0) if the copy was successful or one (1) if any error occurs. You can use this exit code to determine whether or not the RCP was successful if you are using RCP in a batch file (use the IF ERRORLEVEL command to check it) or if you are running RCP from within another program (use the Win32 GetExitCodeProcess API call). An error exit code may be caused by a connection problem (a TCP/IP error) or a file access problem, either locally or remotely (for example, the file does not exist, or permission is denied). If you are copying multiple files, if an error is received while copying **any** of the files, RCP will exit with an error exit code (1). Some of the files may have been successfully copied before the error occurred.

Note: If you have checked the configuration option labeled **Use NT's RCP Syntax**, the exit code if any error occurs will be negative one (**-1)** instead of 1. This is done to mirror the native Windows rcp command.

ASCII DATA CONVERSION

The -a option translates the file regardless of its actual contents. For example, if you transfer a non-text file using -a, it will be modified as if it were a text file. The resulting file will most likely be unusable.

If you use the -a option and you transfer multiple files, all files will be translated as ASCII files.

Using the -a option to transfer files **to** the remote host will slow the operation of Winsock RCP somewhat because it must read each file twice. It reads it once to calculate the new translated file size, then reads it again to transfer the data. This is because the RCP protocol requires that the exact file size be transmitted before the actual data in the file is sent. Without the -a option, the file size can be found by examining the file's directory entry, but with the -a, the file's contents must be examined to determine the size after CR/NL combinations are replaced with NL.

The -a option will also slow Winsock RCP when transferring files *from* the remote host, but only slightly. If transmission speed is critical, consider using utilities to translate the text files after they are transferred.

RCP THROUGH REXEC (RCP/X)

The standard RCP protocol initiates the connection to the remote host use the *rsh* (remote shell) protocol. The *rsh* protcol requires the establishment of *host equivalence* with the remote host so no password is required. As explained in the section on Security, host equivalence can pose security risks.

Winsock RCP supports an alternative method of initiating the connection to the remote host using the *rexec* (remote execution) protocol instead. The difference between the *rsh* and *rexec* protocols is that *rexec* does not require host equivalence. It requires you to specify a valid password for the remote host for each RCP copy.

This method of using the *rexec* protocol instead of *rsh* should work with any Unix system running the *rexecd* daemon. It is possible that some systems may have a problem with using this method since the original Unix implementation of rcp did not allow for this, but it should work with most Unix systems.

EXAMPLES

* To copy a file from the PC to a remote host, use:

```
rcp localfile remhost:/u/joe
```

The file localfile is copied from the PC to the remote host remhost and placed in the directory /u/joe.

* To copy a file from the remote host to the PC, use:

```
rcp remhost:/u/joe/remfile \lists\remfile
```

The file remfile is copied from the remote host remhost to the file remfile in the directory \lists on the PC.

* To copy a remote file from one remote host to another remote host, use:

```
rcp host1:/u/joe/xfile host2:/u/fred/yfile
```

The file /u/joe/xfile on host1 is copied to the file /u/fred/yfile on the remote host host2.

* To copy all of the files in the directory \docs to the a remote host:

```
rcp -a -w \docs\*.* remhost:/u/docs
```

All of the files in \docs are copied to the directory /u/docs on the remote host remhost. The files are converted from the MS-DOS text file format to the Unix text file format using the "-a" option. As the files are transferred, their names will display on the screen. The "-w" option tells Winsock RCP to wait for you to press Enter after the files are transferred so you can examine the list.

* To copy the file from a remote host to the diskette in drive A: on the PC:

```
rcp mary@remhost:resume.doc a:.
```

This will copy the file resume.doc from the user mary's home directory (since no starting directory was given after the host) to the same name on the diskette in drive A: on the PC. Note the "." after the colon of the drive letter. This refers to the current directory on the A: drive, since a destination filename or directory name is required.

* To send the entire directory tree from the PC to a remote host, use:

```
rcp -r \share joe@remhost:
```

The directory \share is copied from the PC to the home directory of joe on the remote host remhost.

* To copy the files beginning with the letters a, b, and c, and ending with .pl from the remote host running Unix to the \perl directory on the PC, use:

```
rcp remhost:/u/perl/[a-c]*.pl d:\perl
```

Since in this case the host is a Unix system and the files are being copied from the Unix system, you can use wildcard characters that are valid at the Unix host. If the files were being copied from the PC to the Unix system, you would only be able to use wildcard characters valid on the PC (* and ? only).

NOTE: You can substitute "wrcp" for all of the above examples if you wish to use the windowing version of Winsock RCP.

WINSOCK RSH

Winsock RSH for Win32 executes a command on a remote host and displays the results on your PC's screen or stores the output in a file. It is similar to the Unix utility of the same name.

Winsock RSH differs from Winsock REXEC in that Winsock RSH does not require a password. It depends on "*host equivalence*", which is explained in the section on Security.

The remote host must be a system running *rshd* (remote shell daemon). This can be a Unix system, a Windows system, or another operating system with a Unix compatible remote shell Daemon. Windows does not come with a remote shell daemon. You can use Denicomp Systems' Winsock RSHD, Winsock RSHD/95, or Winsock RSHD/NT.

There are two versions of Winsock RSH: a "console" version that well-suited for execution from a Windows Command Prompt (named **rsh**) or batch files and a windowing version that displays output in its own window and allows you to use scroll bars to review the output, cut/paste the output, and save the output in a file. The windowing version is named **wrsh**.

Please note that if you are using Windows NT, 2000, or XP, it includes its own rsh. Since the rsh included in this package has the same name, you must be sure to put the directory that contains Winsock RSH at the beginning of your PATH environment variable (or rename the Windows native rsh command).

IMPORTANT! In general, you should not try to use Winsock RSH to execute interactive remote commands (commands that require keyboard input). Its use for this purpose is not supported. Some commands may work through RSH and others may not. You should use *telnet* or *rlogin* for interactive sessions. Remote commands that simply require interaction through the standard input, standard output, and standard error may work interactively through RSH. Programs that require more sophisticated input or output will not. When the remote host is a Unix system for example, the remote shell daemon (rshd) does not associate the process with any psuedo-tty as it does with rlogin or telnet, so some methods of input and output will not work.

SYNTAX

Console Version:

```
rsh [-abcdenovw] [-l User] [-i File] [-r File | -t File]
                 [-s File | -u File] Host { Command | @File }
```

Windowing Version:

```
wrsh [-abcdehmnovw] [-l User] [-i File] [-r File | -t File]
                    [-s File | -u File] Host { Command | @File }
```

PARAMETERS

Host

The host name of the remote host on which the command is to be executed.

Command

The command to execute. If the command contains special characters that are interpreted by a command shell, you must enclose the command inside double quotes (" ").

@File

Instead of specifying the Command on the command line, you can store the command to execute in a file. If this parameter begins with the at-sign (@), the command is read from the filename following it. (Do not put any spaces between the @ and the filename.) Since the length of the command line is limited, you can specify longer commands (up to 8192 characters, assuming the remote host allows commands this long) by storing them in a file.

NOTES ABOUT COMMAND LINE PARAMETERS

The ***Host*** parameter can appear either before or after the command line options.

The ***Host*** may either be a host name or an IP address. If you use a host name, it must be resolvable, either through the HOSTS file or through DNS (Domain Name Service).

Winsock RSH will not expand wildcard characters to match filenames on the local PC, so commands that contain wildcard characters (* and ?) do not need to be enclosed in quotes, unless some other part of the command makes it necessary.

By default, end-of-line conversions are performed on the standard input, standard output, and standard error unless they are redirected to or from a file or pipe; then it is treated as binary data. If the channel is redirected (using either > or < , |, or one of the command line options), the default for that channel is that no end-of-line conversions are performed. This behavior can be modified using command line options.

COMMAND LINE OPTIONS

-a	Perform end-of-line conversions on the standard input, standard output ,and standard error. This converts a line feed (LF), the Unix standard line terminator, to a carriage return (CR) and a line feed, the DOS/Windows standard.
-b	Do not perform end-of-line conversions on the standard input, standard output, and standard error. Data is sent and received unchanged.
-c	Combine the standard output and standard error from the remote command together and send it to the standard output.
-d	Perform end-of-line conversions on the standard input. This is done by default if the standard input is not being redirected from a file or pipe. However, if you are redirecting standard input from a file or pipe and you want the conversion to be done, use this option. It will modify the data before it is sent to the remote command.
-e	Perform end-of-line conversions on the standard error. This is done by default if the standard error is not being redirected to a file or pipe. However, if it is being redirected and you want the conversion to be done, use this option.
-h	Run Hidden. Normally, Winsock RSH will display a window showing the output of the command executed and any possible error messages. With -h, no window will display. However, you will not be able to see any errors that might occur nor will you be easily able to stop the command should it be necessary. This is useful for software developers who wish to transparently call Winsock RSH from within their software.

	This is available in the windowing version (wrsh) only. Also, it **must** appear before the ***Host*** parameter and before any other option that requires an additional argument (such as ***-l User***).
-i File	Sends the contents of ***File*** as the standard input to the remote command. This is the same as using "<". It is provided to allow standard input redirection in the windowing version (wrsh), which does not allow the use of "<". The option is available in both rsh and wrsh, however.
-l User	Specifies that Winsock RSH should log in to the remote host as the user specified instead of the local user name. If this option is not specified, the local user name is determined by Winsock RSH as explained later.
-m	Run Minimized. Normally, Winsock RSH will display a window showing the output of the command executed and any possible error messages. With **-**m, Winsock RSH will only display a minimized icon while running. This is useful for software developers who wish to transparently call Winsock RSH from within their software. This is available in the windowing version (wrsh) only. Also, it **must** appear before the ***Host*** parameter and before any other option that requires an additional argument (such as ***-l User***).
-n	Specifies that the standard input is null. No data will be read from Winsock RSH's standard input and sent to the remote command.
-o	Perform end-of-line conversions on the standard output. This is done by default if the standard output is not being redirected to a file or pipe. However, if it is being redirected and you want the conversion to be done, use this option.
-r File	Redirects the standard output from the remote command to ***File***. This is the same as using ">". It is provided to allow standard output redirection in the windowing version (wrsh), which does not allow the use of ">". The option is available in both rsh and wrsh, however.
-s File	Redirects the standard error from the remote command to ***File***. This is the same as using "2>". It is provided to allow standard error redirection in the windowing version (wrsh), which does not allow the use of "2>". The option is available in both rsh and wrsh, however.
-t File	Redirects the standard output from the remote command to ***File*** and displays the output on the screen.
-u File	Redirects the standard error from the remote command to ***File*** and displays the output on the screen.
-v	Displays the version number of Winsock RSH.
-w	Wait after the command completes. If you are using the console version (rsh), this will cause Winsock RSH to prompt and wait for you to press the Enter key when the command is finished executing. If you are using the windowing version (wrsh), this will wait for you to close the RSH window (using File/Exit from the menu for example). This allows you to use the scroll bars to review the output, cut/paste it, or save it in a file. Without **-**w, the window will close at the end of the command automatically.

USER NAMES

The user name (or login name) determines the file access privileges permitted at the remote host.

If you use the **-**l option, that user name is used at the remote host. If you do not, the local user name is used.

The local user name is normally the name you used when logging in to Windows. For example, if you logged in to Windows as the user "joed", Winsock RSH will use "joed" as the user name at the remote host. Winsock RSH will always convert this name to all lowercase characters.

You can override the Windows login user name when using Winsock RSH by specifying an alternate user in the **R-Commands** applet in the Control Panel. Specify the alternate user either the **RSH – All Users** tab or the **RSH – Current User** tab. The value entered in the **Current User** tab will only be in effect when you log into Windows; it allows you to specify other user names for other users if multiple people use your system.

EXIT CODES

RSH will return an exit code value of zero (0) if it successfully connected to the host and submitted the command for execution or one (1) if any error occurs. You can use this exit code to determine whether or not the RSH was successful if you are using RSH in a batch file (use the IF ERRORLEVEL command to check it) or if you are running RSH from within another program (use the Win32 GetExitCodeProcess API call).
An error exit code may be caused by a connection problem (a TCP/IP error) or permission problem on the host (for example, the user is denied access). However, once the connection is established and the remote shell daemon (rshd) grants permission to execute the command, you will receive a successful exit code (0) even if the command on the remote host fails. The exit code from RSH denotes a successful command submission; it does not indicate the success or failure of that command.

The *rsh* protocol does not provide a method of retrieving the exit code of the remote command. If you need the exit code of the remote command, one way to do this is to echo the exit code to the standard output, then capture the standard output to a file and read the last line of the file for the exit code. For example:

```
rsh unixhost "do_backup; echo $?" > result.txt
```

This will create a file called "result.txt' on your system and the last line of the file will be the exit code of the remote command "do_backup". The above will work if you use the Bourne shell (sh) or Korn shell (ksh). If you use the C-Shell (csh), substitute "echo $status".

EXAMPLES

* To display the users logged in to a remote host, use:

```
wrsh remhost who
```

The list of users is displayed on your PC. As you see, the window is closed when the command ends, so this command is not very useful. Try this:

```
wrsh -w remhost who
```

This will list the users on the remote host, then wait for you to close the window. If the list of users scrolled the window, you can use the scrollbars to view the scrolled information.

* To list the files in the /usr directory ending with .txt on the remote host, use:

```
rsh remhost ls /usr/*.txt
```

The files in the /usr directory are displayed. This example was used from a Command Prompt, so the output was displayed in the Command Prompt window.

* This lists the contents of the /usr directory on the remote host and stores it in the file "usrfiles.txt" on your PC. We will run the command minimized so the RSH window does not display:

```
wrsh -m -r \lists\usrfiles.txt remhost ls /usr
```

-

WINSOCK REXEC

Winsock REXEC for Win32 executes a command on a remote host and displays the results on your PC's screen or stores the output in a file. It is similar to the Unix utility of the same name.

Winsock REXEC differs from Winsock RSH in that it requires you to supply a password; it does not depend on host equivalence as does RSH. The command will not be executed unless you supply the correct password. The password is validated by the remote host system.

Security Note: The rexec protocol sends the user login and password over the network as clear text. This information could be extracted by a “network sniffer” running on your network.

The password can be typed at the keyboard when you execute the command, you can include it in the REXEC command line, or you can specify it in the Winsock RCP/RSH/REXEC **R-Commands** Control Panel applet so you do not need to type it each time.

The remote host must be a system running the *rexecd* server. This can be a Unix system, a Windows system, or another operating system with a Unix compatible rexecd. Windows does not come with an rexecd. You can use Denicomp Systems’ Winsock REXECD/95 or Winsock REXECD/NT.

There are two versions of Winsock REXEC: a "console" version that well-suited for execution from a Command Prompt (named **rexec**) and a windowing version that displays output in its own window and allows you to use scroll bars to review the output, cut/paste the output, and save the output in a file. The windowing version is named **wrexec**.

Please note that if you are using Windows NT, 2000, or XP, it includes its own rexec. Since the rexec included in this package has the same name, you must be sure to put the directory that contains Winsock REXEC at the beginning of your PATH environment variable (or rename the native rexec command).

IMPORTANT! In general, you should not try to use Winsock REXEC to execute interactive remote commands (commands that require keyboard input). Its use for this purpose is not supported. Some commands may work through REXEC and others may not. You should use *telnet* or *rlogin* for interactive sessions. Remote commands that simply require interaction through the standard input, standard output, and standard error may work interactively through REXEC. Programs that require more sophisticated input or output will not. When the remote host is a Unix system for example, the remote exec daemon (rexecd) does not associate the process with any psuedo-tty as it does with rlogin or telnet, so some methods of input and output will not work.

SYNTAX

Console Version:

```
rexec [-abcdenovw] [-l User] [-p Password] [-i File] [-r File | -t File]
                   [-s File | -u File] Host { Command | @File }
```

Windowing Version:

```
wrexec [-abcdehmnovw] [-l User] [-p Password] [-i File] [-r File | -t File]
                      [-s File | -u File] Host { Command | @File }
```

PARAMETERS

Host

The host name of the remote host on which the command is to be executed.

Command

The command to execute. If the command contains special characters that are interpreted by a command shell, you must enclose the command inside double quotes (" ").

NOTE: Winsock REXEC will not expand wildcard characters to match filenames on the local PC, so these do not need to be enclosed in quotes.

@File

Instead of specifying the Command on the command line, you can store the command to execute in a file. If this parameter begins with the at-sign (@), the command is read from the filename following it. (Do not put any spaces between the @ and the filename.) Since the length of the command line is limited, you can specify longer commands (up to 8192 characters, assuming the remote host allows commands this long) by storing them in a file.

NOTES ABOUT PARAMETERS

The ***Host*** parameter can appear either before or after the command line options.

The ***Host*** can be either a host name or IP address. If you use a host name, it must be resolvable, either through the HOSTS file or through DNS (Domain Name Service).

Winsock REXEC will not expand wildcard characters to match filenames on the local PC, so commands that contain wildcard characters (* and ?) do not need to be enclosed in quotes, unless some other part of the command makes it necessary.

By default, end-of-line conversions are performed on the standard input, standard output, and standard error unless they are redirected to or from a file or pipe; then it is treated as binary data. If the channel is redirected (using either > or <, |, or one of the command line options), the default for that channel is that no end-of-line conversions are performed. This behavior can be modified using command line options.

COMMAND LINE OPTIONS

-a	Perform end-of-line conversions on the standard input, standard output ,and standard error. This converts a line feed (LF), the Unix standard line terminator, to a carriage return (CR) and a line feed, the DOS/Windows standard.
-b	Do not perform end-of-line conversions on the standard input, standard output, and standard error. Data is sent and received unchanged.
-c	Combine the standard output and standard error from the remote command together and send it to the standard output.
-d	Perform end-of-line conversions on the standard input. This is done by default if the standard input is not being redirected from a file or pipe. However, if you are redirecting standard input from a file or pipe and you want the conversion to be done, use this option. It will modify the data before it is sent to the remote command.
-e	Perform end-of-line conversions on the standard error. This is done by default if the standard error is not being redirected to a file or pipe. However, if it is being redirected and you want the conversion to be done, use this option.

-h	Run Hidden. Normally, Winsock REXEC will display a window showing the output of the command executed and any possible error messages. With -h, no window will display. However, you will not be able to see any errors that might occur nor will you be easily able to stop the command should it be necessary. This is useful for software developers who wish to transparently call Winsock REXEC from within their software. This is available in the windowing version (wrexec) only. Also, it **must** appear before the ***Host*** parameter and before any other option that requires an additional argument (such as ***-l User***).
-i File	Sends the contents of ***File*** as the standard input to the remote command. This is the same as using "<". It is provided to allow standard input redirection in the windowing version (wrexec), which does not allow the use of "<". The option is available in both rexec and wrexec, however.
-l User	Specifies that Winsock REXEC should log in to the remote host as the user specified instead of the local user name. If this option is not specified, the local user name is determined by Winsock REXEC as explained later.
-m	Run Minimized. Normally, Winsock REXEC will display a window showing the output of the command executed and any possible error messages. With -m, Winsock REXEC will only display a minimized icon while running. This is useful for software developers who wish to transparently call Winsock REXEC from within their software. This is available in the windowing version (wrexec) only. Also, it **must** appear before the ***Host*** parameter and before any other option that requires an additional argument (such as ***-l User***)..
-n	Specifies that the standard input is null. No data will be read from Winsock REXEC's standard input and sent to the remote command.
-o	Perform end-of-line conversions on the standard output. This is done by default if the standard output is not being redirected to a file or pipe. However, if it is being redirected and you want the conversion to be done, use this option.
-p Password	Specifies the password of the user to be used at the remote host. If you do not specify the password on the command line or store it using the Control Panel, Winsock REXEC will require you to enter it when you execute the command.
-r File	Redirects the standard output from the remote command to ***File***. This is the same as using ">". It is provided to allow standard output redirection in the windowing version (wrexec), which does not allow the use of ">". The option is available in both rexec and wrexec, however.
-s File	Redirects the standard error from the remote command to ***File***. This is the same as using "2>". It is provided to allow standard error redirection in the windowing version (wrexec), which does not allow the use of "2>". The option is available in both rexec and wrexec, however.
-t File	Redirects the standard output from the remote command to ***File*** and displays the output on the screen.
-u File	Redirects the standard error from the remote command to ***File*** and displays the output on the screen.
-v	Displays the version number of Winsock REXEC.
-w	Wait after the command completes. If you are using the console version (rexec), this will cause Winsock REXEC to prompt and wait for you to press the Enter key when the command is finished executing. If you are using the windowing version (wrexec), this will wait for you to close the REXEC window (using File/Exit from the menu for example). This allows you to use the scroll bars to review the output, cut/paste it, or save it in a file. Without -w, the window will close at the end of the command automatically.

USER NAMES

The user name (or login name) determines the file access privileges permitted at the remote host.

If you use the **-**l option, that user name is used at the remote host. If you do not, the local user name is used.

The local user name is normally the name you used when logging in to Windows. For example, if you logged in to Windows as the user "joed", Winsock REXEC will use "joed" as the user name at the remote host. Winsock REXEC will always convert this name to all lowercase characters.

You can override the Windows login user name when using Winsock REXEC by specifying an alternate user in the **R-Commands** applet in the Control Panel. Specify the alternate user either the **REXEC – All Users** tab or the **RSH – Current User** tab. The value entered in the **Current User** tab will only be in effect when you log into Windows; it allows you to specify other user names for other users if multiple people use your system. If you use the **-**l option, the user name specified after it on the command line is used at the remote host. If you do not specify the **-**l option, the local user name is used at the remote host.

EXIT CODES

REXEC will return an exit code value of zero (0) if it successfully connected to the host and submitted the command for execution or one (1) if any error occurs. You can use this exit code to determine whether or not the RSH was successful if you are using REXEC in a batch file (use the IF ERRORLEVEL command to check it) or if you are running REXEC from within another program (use the Win32 GetExitCodeProcess API call).

An error exit code may be caused by a connection problem (a TCP/IP error) or permission problem on the host (for example, the password is invalid). However, once the connection is established and the remote exec daemon (rexecd) grants permission to execute the command, you will receive a successful exit code (0) even if the command on the remote host fails. The exit code from REXEC denotes a successful command submission; it does not indicate the success or failure of that command.

The *rexec* protocol does not provide a method of retrieving the exit code of the remote command. If you need the exit code of the remote command, one way to do this is to echo the exit code to the standard output, then capture the standard output to a file and read the last line of the file for the exit code. For example:

```
rexec unixhost "do_backup; echo $?" > result.txt
```

This will create a file called "result.txt' on your system and the last line of the file will be the exit code of the remote command "do_backup". The above will work if you use the Bourne shell (sh) or Korn shell (ksh). If you use the C-Shell (csh), substitute "echo $status".

EXAMPLES

* To display the users logged in to a remote host, use:

```
wrexec remhost who
```

This will first ask you for your password if it is not stored in WIN.INI. The list of users is then displayed in the REXEC window on your PC. As you see, the window is closed when the command ends, so this command is not very useful. Try this:

```
wrexec -w remhost who
```

-

This will list the users on the remote host, then wait for you to close the window. If the list of users scrolled the window, you can use the scrollbars to view the scrolled information.

* To list the files in the /usr directory ending with .txt on the remote host, use:

```
rexec -l bob -p xyzxyz -w remhost ls /usr/*.txt
```

The files in the /usr directory are displayed in console window. Again, the -w option is used to wait for the Return key so you can see the results. Also, the user is specified with the -l option and the password is specified after the -p option, so Winsock REXEC will not ask you to type it in before the command executes.

SUPPORT

Support is available via e-mail.

Internet: support@denicomp.com
WWW: http://www.denicomp.com