

rsh

remote shell

SYNOPSIS

```
rsh [username@]hostname [-DNn] [-l username] command
rsh [-DNn] [-l username] [username@]hostname command
```

DESCRIPTION

rsh executes *command* on the specified *hostname*. To do so, it must connect to a [rshd](#) service (or daemon) on the *hostname* machine.

When no user name is specified either with the `-l` option or as part of *username@hostname*, **rsh** connects as the user you are currently logged in as. If this is a domain user (that is, of the form *domain\username*), the whole name is sent. If you are not logged in as a domain user, the name *machine_name\username* is sent.

For example, if you logged in as the domain user *mydomain\myname*, the command

```
rsh otherhost pwd
```

runs the [pwd](#) command as user *mydomain\myname* on the remote machine *otherhost*, as do the commands:

```
rsh otherhost -l mydomain/myname pwd
rsh otherhost -l mydomain\myname pwd
rsh mydomain/myname@otherhost pwd
rsh mydomain\myname@otherhost pwd
```

If an attempted **rsh** connection to the host fails because of a "permission denied" or "remuser too long" error, the utility immediately tries again, sending only the user name without domain (or machine) information.

Note:

Because the **rsh** and [rcp](#) utilities resend the current without the domain if it is too long and the [rlogin](#) utility does not, a user may require two entries in the `hosts.equiv` or `.rhosts` file. If the full name (including domain) is too long for the [rshd](#) service (or daemon) being used, the user needs one entry with the full user name (including domain) for use with [rlogind](#) and a second with the the same user name minus the domain for use with [rshd](#).

rsh actually sends two user names to the [rshd](#) service (or daemon), `remuser` and `locuser`.

`remuser` is your user name that you are currently logged into the client machine as (and includes domain information if you logged in as a domain user). It is called `remuser` by the service (or daemon) because from the point of view of the service (or daemon), the client machine is remote. `remuser` is the name that must appear in either the global `hosts.equiv` file or the appropriate `.rhosts` file on the server. `remuser` cannot be set by the user.

`locuser` is the user name that the service (or daemon) uses to execute the command on the server. Again, it is

named as such, because from the point of view of the service (or daemon), the server is the local machine. This is either the user name that you are currently logged in as (minus any domain information if you are a domain user) or the user name explicitly entered on the **rsh** command line.

rsh copies its standard input to the remote command, the standard output of the remote command to its standard output, and the standard error of the remote command to its standard error (unless **-N** is specified).

Non-quoted shell metacharacters on the command line are interpreted on the local machine, while quoted metacharacters are interpreted on the remote machine. For example, the command

```
rsh otherhost cat remotefile >> localfile
```

appends the remote file *remotefile* to the local file *localfile*, while

```
rsh otherhost cat remotefile \>\> other_remotefile
```

appends *remotefile* to *other_remotefile*.

rsh normally terminates when the remote command does.

Options

-D

enables socket debugging on the TCP sockets used for communication with the remote host. This option also displays the `locuser` and `remuser` being sent to [rshd](#).

Note:

Because **rsh -D** is looking at the world from the client machine's point of view (as opposed to the host machine's point of view), the name displayed as `local_user` is actually the one sent as `remuser` and the name displayed as `remote_user` is actually `locuser`.

This option is useful for determining the exact case of your domain and user names, which must be correct in the `.rhosts` and `hosts.equiv` files. Also if the `local_user` name contains a backslash (`\`), you are attempting to connect as a domain user.

-l username

specifies the remote user name to be used when executing the command. This can also be specified by including `username@hostname` on the command line.

-N

does not generate a separate standard error stream. All output is sent to standard output. This is useful when you are running an interactive command. For example,

```
rsh -N localhost cmd
rsh -N localhost sh -i
```

-n

redirects input from the special device `/dev/nul`.

Troubleshooting

If you are having trouble connecting to [rshd](#) on a remote host, here are some possible solutions to your problems.

If you are receiving a "permission denied" error, it could be for one of the following reasons:

- There is no entry for the `remuser` on the remote host in either the global `hosts.equiv` file or the `locuser`'s `.rhosts` file. Check these files to ensure that an entry exists for `remuser` on the machine you are connecting from. You can use `rsh -D` to determine the names being sent as `locuser` and `remuser`.

Note:

UNIX systems may not always know your Windows machine by the same name that other Windows machines do. You can attempt to find out the name by which a UNIX system knows your machine by using the [rlogin](#) command to connect to that UNIX system and typing

```
/bin/who am i
```

This should display the name of your machine in parentheses at the end of the information line. If you cannot find the machine name using this method, consult the UNIX system's administrator.

- A `hosts.equiv` was not found in the **ROOTDIR**/`etc` directory on a remote host running the MKS Toolkit [rshd](#) service or in the `/etc` directory on remote UNIX host.
- A `.rhosts` was not found in the home directory for `locuser` on the remote host, normally indicated by the setting of the **HOME** environment variable for that user. [rshd](#) runs programs in this directory. On remote hosts running MKS Toolkit [rshd](#), you cannot depend on the `locuser`'s **HOME** environment variable correctly identifying that user's home directory.

By default, **HOME** is set to `%HOMEDRIVE%%HOMEPATH%`. However, the **HOMEDRIVE** and **HOMEPATH** environment variables are only available to interactive applications and, thus, not available to services. If `locuser`'s **HOME** environment variable is set to the default or is defined using **HOMEDRIVE** or **HOMEPATH**, [rshd](#) cannot use **HOME** to identify the user's home directory. In such a case, [rshd](#) uses a reasonable default as the home directory.

For this reason, it is recommended that the **HOME** environment variable be explicitly set (without using **HOMEDRIVE** or **HOMEPATH**) for each user that [rshd](#) can be run as. For instructions on setting and viewing environment variables, see your Operating System's online help.

To determine the home directory of `locuser` on the remote host, type

```
rexec hostname -l username pwd
```

where *hostname* is the remote host and *username* is `locuser` on that machine. You are prompted for `locuser`'s password and the home directory for `locuser` is then displayed. This is the directory where the `.rhosts` file should reside.

If you are receiving a "could not retrieve password" error when attempting to connect to a remote host running the MKS Toolkit [rshd](#) service, it most likely means that the [rsetup](#) command has not been used to create an entry in the LSA database for `locuser`.

If you are receiving `connection to server failed` error, it usually implies that either there is no [rshd](#) service (or daemon) running on the remote host or that there were more concurrent `rsh` requests than the remote host could handle.

Most versions of [rshd](#) have a limit to the length of user name they can handle. On some UNIX systems, this limit is as low as 14 characters. As a result, you may receive an error message stating that either `remuser` or `locuser` is too long.

When `remuser` is too long, you need a shorter user name on the client machine.

When the "locuser is too long" error is received, it means that not only is the `locuser` name longer than the server's limit, but, by extension, it is an invalid user name on that server.

DIAGNOSTICS

Possible exit status values are:

0
Successful completion.

>0
An error occurred.

ACKNOWLEDGEMENT

This product includes software developed by the University of California, Berkeley and its contributors.

PORTABILITY

All UNIX systems. Windows Me. Windows NT 4.0. Windows 2000. Windows XP. Windows Server 2003.

NOTE

When you are connected to [rshd](#) running on a Windows machine, it takes longer to run a command as a domain user than it does to run that same command as a local user. This extra time is the result of the server having to contact the network for the additional domain information.

AVAILABILITY

MKS Toolkit for System Administrators

MKS Toolkit for Developers
MKS Toolkit for Interoperability
MKS Toolkit for Professional Developers
MKS Toolkit for Enterprise Developers
MKS Toolkit for Enterprise Developers 64-Bit Edition

SEE ALSO

Commands:

[rcp](#), [rexec](#), [rlogin](#), [rsetup](#), [rshd](#)

File Formats:

[hosts.equiv](#), [rhosts](#)

MKS Toolkit Connectivity Solutions Guide

MKS Toolkit 9.0 Documentation Build 78.