

Remote Executor

This chapter describes how to use the Cisco IOS for S/390 Remote Executor. It includes these sections:

- **What Is the Remote Executor?**
A brief description of the Remote Executor, TCPREXEC.
- **The REMCMND Program**
Describes how the REMCMND program passes information to the REXEC daemon.
- **Running REMCMND Interactively (TCPREXEC)**
Describes how to run the REMCMND program from a TSO command line with the TCPREXEC command.
- **Running REMCMND In Batch Mode**
Describes how to run the remote executor in batch mode. Example JCL is provided.

What Is the Remote Executor?

The Remote Executor, TCPREXEC, uses the REMCMND program to execute a command on a foreign host and receive results on the local host. You can use the TSO command TCPREXEC or you can run TCPREXEC in batch mode using a JCL.

This product uses version 5.5 of SAS C.

The REMCMND Program

The REMCMND program lets you execute a command on a remote host and receive results on the local host. The REMCMND program can be executed from the TSO command line or from a batch job using a JCL.

An REXEC daemon must be running on the host. The REMCMND client passes the user name, password, and command to the REXEC daemon. The daemon provides logon and user authentication, depending on the parameters set by the user. If authentication fails, an error message displays on the local host.

Running REMCMND Interactively (TCPREXEC)

When executing REMCMND interactively under TSO, you must invoke a REXX EXEC called TCPREXEC to dynamically allocate statements and files. TCPREXEC passes command-line arguments to the host in as-is format (overriding the TSO function which makes all arguments uppercase). The TCPREXEC REXX EXEC is contained in the SAMP library. Copy it to a system CLIST or REXX library to use it.

Note Information can be passed to **REMCMND** by either the **PARM** string or the **SYSIN DD** file. However, data cannot be entered both on the **PARM** string and in the **SYSIN DD** file.

To execute a command on the foreign host and receive the results on the local host, use the **TCPREXEC** command as shown here:

```
TCPREXEC ' {USER (username)}
          {PASS (password)}
          [{SUBSYS (subsysid)}]
          hostname
          command_string '
```

Syntax Description

USER	Specifies the user ID on the foreign host. The default user name (if you wait for the prompt) is your TSO user ID, in upper case. If your user ID is the same on the host system but in lower case, it must be retyped (in lower case). If this parameter is not supplied, you will be prompted for it.
PASS	Specifies the password of the user ID on the foreign host. Do not specify PASS() on the command line if no password is used. If prompted for a password, just press the enter key. If this parameter is not supplied, you will be prompted for it.
SUBSYS	Specifies a particular Cisco IOS for S/390 gateway rather than the default ACSS. This parameter is optional.
<i>hostname</i>	The name of the foreign host that is the target of the REMCMND command. <i>hostname</i> can be a fully qualified name or alias, or the IP net address, in the form nnn.nnn.nnn.nnn. This parameter is required.
<i>command_string</i>	The command sent to the foreign host, composed of one or more words. The code checks for the user name, password, subsystem ID, and host name, then assigns the remaining string as the command.

Using the NETRC File with TCPREXEC

REMCMDND uses a NETRC (user.FTP.NETRC on MVS) file if you provide one. The NETRC file contains the host name, user ID, and password for multiple machines on your network. It takes this format:

MACHINE *hostname*

LOGIN *username*

PASSWORD *password*

Use this sequence of MACHINE, LOGIN, PASSWORD for each of the hosts you want to connect to.

TSO prepends the current local user ID to FTP.NETRC, then searches the MVS catalog for that file. When TCPREXEC finds user.FTP.NETRC, it reads that data set, looking for the machine name you requested (hostname). When a match is found TCPREXEC reads the login and password statements and uses those for authentication to the remote host.

Note When a NETRC file is used, the case of the machine name, login name, and password must match the case of the name used in the **SYSIN**.

Example:

To have the **TCPREXEC** program scan your NETRC file for a machine with the host name snoopy, enter this **TCPREXEC** command:

TCPREXEC snoopy ls -l

In this command, snoopy is the host name and **ls -l** is the command string.

If TCPREXEC does not find user.FTP.NETRC, or does not find the host name in the file, it prompts for the user name and password.

Note FTP2 uses the same NETRC file that TCPREXEC uses. Read Using the NETRC File with TCPREXEC for more information.

Running REMCMDND In Batch Mode

To run REMCMDND in batch mode, you must include information in a JCL stream.

If you want to run in batch mode, use the JCL in this example. This example JCL issues an **ls -l** command on the remote host *hostname*.

```
//REMCMDND JOB
//JS01 EXEC PGM=REMCMDND,
// PARM='USER(username) PASS(password) hostname ls -l'
//STEPLIB DD DISP=SHR,DSN=T01TCP.LINK
// DD DISP=SHR,DSN=T01TCP.LOAD
//SYSPRINT DD SYSOUT=A
//SYSTEM DD SYSOUT=A
//OUTPUT DD SYSOUT=A
```

Output received from the command executed by program REMCMDND is written to the OUTPUT DD statement.

Using a NETRC file in Batch Mode

When you run REMCMND in batch mode, REMCMND does not automatically read your NETRC file as it does for the interactive TSO command TCPREXEC. In batch mode, you must include a NETRC statement in your JCL.

The next example uses the user.FTP.NETRC file to provide the user name and password.

```
//RECMND      JOB
//JS01 EXEC  PGM=RECMND,PARM='hostname ls -l'
//STEPLIB    DD DISP=SHR,DSN=T01TCP.LINK
//           DD DISP=SHR,DSN=T01TCP.LOAD
//NETRC      DD DSN=uid.FTP.NETRC,DISP=SHR
//SYSPRINT   DD SYSOUT=A
//SYSTEM     DD  SYSOUT=A
//OUTPUT     DD  SYSOUT=A
```

Note FTP2 uses the userid.FTP.NETRC file. Both FTP2 and REMCMND can use the same NETRC file. Read Client FTP2 in Client FTP2 and Using the NETRC File with TCPREXEC for more information about the NETRC file.

Command Line Arguments

You can use the PARM option on the EXEC statement to pass information to the REMCMND program.

```
//JS01 EXEC  PGM=RECMND,PARM='host_name cmd_string'
```

Combining Commands

On many systems, you can string several commands together on the same line. For example, on UNIX, you can issue two commands on the same line by separating the commands with the semicolon (;) as a delimiter.

Examples

```
% ls ; date
```

You can use this facility with REMCMND in batch files.

```
//JS01 EXEC  PGM=RECMND,PARM='hostname ls;date'
```

Note Because the TSO parser uses the semicolon (;) delimiter for parsing TSO commands, you cannot embed the semicolon in the command string when using the **TCPREXEC** TSO command.

Using a SYSIN DD Statement In Your JCL

If the parameter field on the EXEC statement is blank, REMCMND looks for a SYSIN DD statement in the batch JCL. REMCMND reads an 80 byte card from SYSIN and parses it like the command-line arguments.

Note The SYSIN file is read only if the job is running in batch mode.

Include information for *username*, *password*, *hostname*, and *command_string* in the JCL as shown in this example:

```
//RECMND JOB
//JS01 EXEC PGM=RECMND
//STEPLIB DD DISP=SHR,DSN=T01TCP.LINK
// DD DISP=SHR,DSN=T01TCP.LOAD
//SYSPRINT DD SYSOUT=A
//SYSTEM DD SYSOUT=A
//OUTPUT DD SYSOUT=A
//SYSIN DD *
user_name password host_name cmd_string
//
```

Note Since your parameters can be specified in the **SYSIN** file, the **PARM** field on the **EXEC** statement is not needed.

For frequently used commands, you can include this line in the JCL to create a control file with the pertinent information:

```
USER(user_name) PASS(password) zulu exec /usr/openwin/bin/xioctetn \
-server telnet -host mvshost -title RECMND
```

Using the \ Character for Continuing Long Lines

The backslash character (\) is used to continue a line of text that does not fit on the screen. The backslash may appear anywhere on the line but must be followed by at least one blank space.



Caution Due to parsing limitations, the input statement specified via **SYSIN** must *not* contain sequence numbers in columns 72 through 80, since it will be sent as data to the remote host and cause an error.

