Using TCP/IP Facilities

for MVS Job Submission and Output Retrieval

By Stephen Force

Due to the interest in linking disparate operating systems together, Transmission Control Protocol/Internet Protocol (TCP/IP) is becoming more prevalent in the MVS computing environment. People who previously needed to just submit jobs and examine job output on a MVS host (server) can now do both without leaving their host environment (client) and without logging on to TSO.

This article illustrates two possibilities of doing just this. The first way uses the native functions of MVS TCP/IP File Transfer Protocol (FTP). The second possibility involves some programming that can be implemented fairly easily by experienced TCP/IP socket programmers.

Although some knowledge of TCP/IP would be helpful to understand the concepts presented here, anyone who has ever worked with native TSO or who understands the JES subsystem interface will have no problem. Other readers who desire more information should read the IBM TSO and JCL manuals on job submission, feeling safe that this newly acquired knowledge could help you understand MVS just a tiny bit more. Figure 1 illustrates a simple client to server MVS TCP/IP connection.

FTP/JES Interface

FTP uses the client and server model to offload work from the mainframe (server) to the client, which reduces the demands on the mainframe.

The FTP Server provides four JES interface functions:

- submitting a job;
- displaying the status of all the user's jobs;
- receiving the spool output of the job (JCL messages and SYSOUT); and
- deleting a job.

Submitting a Job

The user creates a jobstream using a text editor on her/his client. The jobname in the JCL must be USERIDn, where n is a one-character letter or number. This job name follows the JES naming conventions as documented. [The TSO/JES subsystem interface does not allow unauthorized users to retrieve output from jobs other than one following the aforementioned naming convention. However, for those of us who need access to these jobs, TSO exit IKJEFF53, JES3 exit IATUX30 and JES2 exit 22 can be written/modified to allow this access.]

Start a session with the target MVS FTP server from your client.

After entering your user ID and password, specify that you want to interface to JES with the Site parameter by typing the following FTP command:

```
SITE FILETYPE=JES
```

To submit the MVS job that you have created, type the following FTP command:

```
PUT filename
```

The JCL is then submitted to the JES internal reader and waits for job initiation. The job is submitted under the user ID that you used when you logged on to the MVS system through FTP.

Display the Status of a Job

After you have submitted the MVS job, you may check the job status:

- waiting for execution;
- executing (running); and
- job has ended.

To do this, issue either the "DIR" or "LS" FTP subcommands while in the FILETYPE=JES mode. These subcommands display the status of all the jobs that are on the JES spool for your user ID.

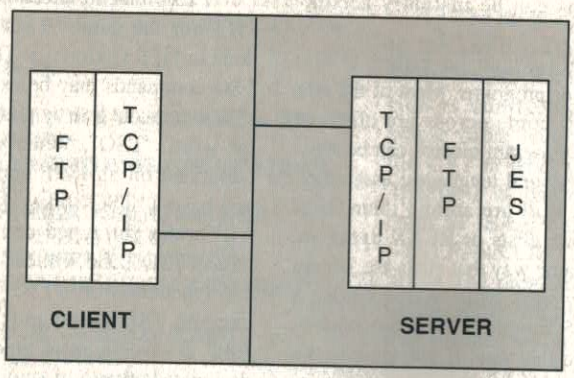
Retrieving Job Output

After the job has completed, you can retrieve the resultant job output files. These files are retrieved one at a time. To be able to obtain job output, the job output must be in a hold queue (JES2) or in a class reserved for external writers (JES3).

To retrieve the job output, use the FTP GET subcommand while in the FILETYPE=JES mode. You must specify both the job ID and the number of the spool file that you want.

For example, to obtain the first job

•• FIGURE 1: Client to Server MVS TCP/IP Connection



output file of job number 1234, issue the FTP subcommand:

```
GET JOB01234.1 JOB1234.out1
```

This subcommand will obtain the first file from job number 1234 and then place the results into client file JOB1234.out1. The client can then work with the retrieved file data.

To obtain all job output files, you can write a procedure (REXX, CLIST, BAT, program, etc.) that extracts each file from JES (using the procedure illustrated previously) and places these data into a single output file. This procedure can run on either the client or the server, depending on the individual needs.

Deleting the Job

As with TSO, a FTP client may delete her/his job before execution or can delete the output of her/his job. This is accomplished by using the DELETE subcommand while in the FILETYPE=JES mode.

For example:

```
DELETE JOB01234
```

The MVS returns the message CANCEL SUCCESSFUL after it deletes the job.

Returning From JES

When you want your FTP client to stop communicating with JES, issue the following FTP subcommand:

```
SITE FILETYPE=SEQ
```

This returns the FTP client to the normal FTP mode.

Automating FTP Functions

Most FTP subcommands can be automated by placing them into a TSO CLIST, REXX EXEC (either on MVS server, or on the client), DOS *.BAT file, etc. These can simplify the FTP use by automating the FTP command issuance.

Some TCP/IP products (such as FTP Software, Inc.'s TCP/IP for DOS) provide commands to help alleviate repetitive commands. For FTP Software, Inc.'s TCP/IP for DOS, the command is:

```
TAKE local_filename
```

where local_filename is a DOS file containing commands.

Job Submit Socket

If you want even more functions (e.g., RACF password prompting, job control pre-checking, job control construction from a skeleton file, etc.), a TCP/IP submit socket could be written.

A socket can be thought of as applications written for TCP/IP systems. Sockets are generally written in the "C" language, but they can also be written in PASCAL.

Generally, a socket has two parts: the client (calling) and the server (called). In a job submit socket scenario, the client would be the PC or workstation. The server would be running on MVS.

An example submit socket could construct a job stream based on desired parameters obtained from the server, then submit the job to MVS through an internal reader (INTRDR). The job stream JCL skeleton could be either hard coded into the server or obtained from a MVS data set allocated to the server.

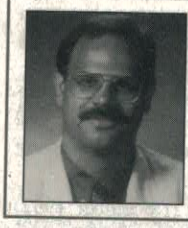
The client would contact the server. The server would obtain the necessary information from the client, and then submit the job to MVS through a internal reader (INTRDR).

Summary

For non-TSO users who need to regularly submit and retrieve jobs from an MVS operating system, and are using systems with TCP/IP installed, this facility might just be for you.

Implementing this facility requires some JES system programmer coordination. Job output hold queue (or external writer for JES3) classes need to be established; JES exits might need to be modified.

This article introduced the reader to the FTP and socket MVS job manipulation facilities. For more information, read the sub-chapter on the FTP/JES interface in the *IBM TCP/IP V2 for MVS: Users Guide.*



Stephen Force is an independent consultant for Chrysler Technology Center in Auburn Hills, Mich.

/*

Was this article of value to you? If so, please let us know by circling Reader Service No. 87.

WIN a FREE NaSPA VIP Tape!

Following are the names of the winners of the December drawing for free NaSPA VIP Tapes. These recipients became eligible for this drawing by using the ***Technical Support*** reader service card to evaluate our editorial, or to enquire about products or services mentioned in the magazine.

Winners receive a copy of the NaSPA VIP Tape, which includes either the MVS or VM/VSE public domain software tape, that is filled with tools and utilities that you can use!

To become eligible for this drawing, just fill out the reader service card located between page 40 and 41, circle the items of interest and drop your card in the mail. The five winners for the February drawing will be randomly selected from reader service cards that reach us by February 28, 1993.

December's Winners

Robert W. Reinke
Anthony DePeppo
William H. Henderson
Ernest Johnson
Tom Simons