CHAPTER 20

TCP/IP and the Web

In today's Internet-connected world, you need to understand more than how to simply turn on your computer and fire up your browser. You also need to understand the basics of using TCP/IP and its associated services. And once you learn the basics, you'll then need to learn how to get "behind the scenes" of the TCP/IP world. You'll also need to learn all about TCP/IP services on your iSeries 400. And that's not always an easy thing to do, because, let's face it, working with the iSeries TCP/IP services isn't easy. There's a lot of information and a lot of complexity built into these services. It's more than one person can know.

That's where this chapter on TCP/IP can help. This chapter offers the best of the combined knowledge of our technical editors and our readers so that you can gain the most complete understanding of how to use your iSeries system and its TCP/IP services in the most effective way possible. With that in mind, you'll find a whole wealth of tips on such varied topics as:

- Transferring iSeries Save Files (*SAVF) with TCP/IP.
- Working with HTTP Server.
- Finding out which port a TCP/IP service is running on.
- Learning to use the QSYS.LIB file system in ways you never thought possible.
- Using SNMP and SMTP.
- Emailing files using the Send Distribution command.
- And much, much more.

So get ready to learn the TCP/IP tips and techniques that will make you the TCP/IP star in your shop.

—Shannon O'Donnell

TCP/IP
and the Web

✦ ✦ ✦ ✦

In This Chapter

APIs

Batch FTP

Email

FTP and *SAVF

FTP and Commands

FTP and IFS

FTP Utility

FTP

HTTP

IFS

PC Commands

SNMP

TCP/IP Address

TCP/IP Communication

TCP/IP Maintenance

TCP/IP Services

✦ ✦ ✦ ✦

APIs

API Really Means "Apply Patch Immediately"

The AS/400's HTTP server provides an excellent way to gather information from your customers and suppliers. To make this task easier, IBM provides several APIs that allow you to process data entered on a form on a Web page with the languages most familiar to AS/400 programmers, such as RPG and COBOL. Unfortunately, if you have upgraded to V4R1 or higher, you have probably discovered that the Convert to Database (QtmhCvtDb) API no longer processes form data correctly.

QtmhCvtDb is used in conjunction with another API, Standard Input (QtmhStdIn), to map data entered into a Web page form to fields in a physical file. Figure 20.1 shows the code for a simple form that allows a customer to enter name and address information in order to be added to a mailing list that is stored in a physical file. (For this example, Name and Address serve as the name and address.) Clicking on the Process Form button initiates a call to the CGI program TESTRPG001. (This is specified as the action on the form tag.)

```
<html>
<head>
<title>QtmhCvtDb API Fix</title>
</head>
<body>
<form METHOD="post" action="/cgi-bin/testrpg001.pgm"">
<div align="center">
<center>
<p><strong>QtmhCvtDb API Fix</strong></p>
</center>
<hr>
<table>
<tr>
<td>Name</td>
<td><input type="text" name="name" SIZE="30"></td>
</tr>
<tr>
<td>Address</td>
<td><input type="text" name="address" size="40"></td>
</tr>
<tr>
<td colspan="2"><input type="submit" value="Process Form"></td>
</tr>
</table>
</div><hr>
<!– The following line of code solves the problem –>
<input type="hidden" name="junk">
</form>
</body>
</html>
```

Figure 20.1: Use this HTML form to allow entry of a name and an address.

A call to QTMHSTDIN retrieves the data from the standard input buffer of the HTTP server and returns the following string to the CGI program:

```
name=Anyone&address=Anywhere
```

The QTMHCVTDB API is then called to map the data entered into the corresponding fields of a physical file. At this point, the physical file field **name** should have a value of **Anyone** and the field **address** should have a value of **Anywhere**. Unfortunately, QTMHCVTDB contains a bug that causes corruption of the data entered in the last form element. If the last form element has a size of one, the data entered is lost entirely; if the form element has a size other than one, the value of the address is returned as a string containing the data entered followed by garbage. Therefore, in this example, address is returned as **Anywhere** plus 32 indecipherable characters that do not look very nice on a mailing label!

Fortunately, there is a simple way to overcome this obstacle. Simply add the following line of code as the last element of the HTML table:

```
<input type="hidden" name="junk">
```

This code causes the corrupted data to be parsed into the hidden field, where it no longer poses a problem. Until a PTF is announced to correct the situation, using this method provides a simple fix.

—David Mayle

Batch FTP

FTP Batch Transfer from AS/400 to PC

Are you looking for a way to get data from your AS/400 to your PC in batch mode? Well, if your PC is attached to your AS/400 via TCP/IP or even a shared network, you can do it. The code shown in Figure 20.2 is a batch program named MYFTP. BAT. MYFTP. BAT uses the FTP switch **-s** to indicate that the batch program—the PC file extension of *. BAT—uses another file (MYFTP.CMD) for FTP commands.

```
FTP -s:C:\windows\myftp.cmd
```

Figure 20.2: Use this PC batch program to download data from your AS/400.

Now, look at the code shown in Figure 20.3. This file contains the commands to connect to my AS/400. The second and third lines are my user ID and password. The FTP command **get** is used to download the file from the AS/400 to the PC. In this case, it downloads a file named CFM05P from the library named SCHEDULE and places that file on my PC's hard drive in a file named CFM05P.TXT. When the FTP program finishes downloading, it ends via the FTP command **quit**. Notice that all the commands are in lowercase. This distinction is important because FTP is an import from UNIX, and UNIX is case-sensitive.

```
open 100.59.191.50
MYUSERID
MYPASSWORD
get SCHEDULE/CFM05P C:\CFM05P.TXT
quit
```

Figure 20.3: These FTP commands connect to and download AS/400 data.

Adding this batch program to an application's macro or even to a scheduling system on your PC is now a simple matter. This method allows your users to download data without needing to run to you for help every time.

—Shannon O'Donnell
—Bradley V. Stone

Email

TCP/IP and MIME

Q: Our company is new to the world of TCP/IP. Lately, in discussions with other TCP/IP folks, I've been hearing about something called MIME. Can you give me a brief explanation of MIME?

A: Multipurpose Internet Mail Extension (MIME) is a compound document messaging standard for the Internet. MIME allows users to transmit multimedia email. This email could include graphics, video, audio, and text. Rich text information, which defines the layout, fonts, and format of email documents, may also be included.

—Kris Neely

Emailing Messages and Files with the SNDDST Command

The Send Distribution (SNDDST) command enables you to send messages and files by email from a CL program or command line to non-AS/400 systems. You can send various types of information or objects with SNDDST. This tip explains three types of information and objects you can send: Long Message (*LMSG), File (*FILE), and Document (*DOC). The *LMSG type allows you to send a message of up to 5000 characters. The only place I've seen documentation for *LMSG is on the cover letters for the PTFs. *LMSG has only two formatting commands, :/N and :/P. The :/N command creates a new line, while :/P creates a new paragraph (which means a blank line and a new line). Here is an example of a SNDDST command using *LMSG and how the received email would look:

```
SNDDST TYPE(*LMSG) TOINTNET((murphyfa@uiuc.edu)) DSTD('Test Message') +
LONGMSG('This is a test message.:/NA new line.:/PA new paragraph.') +
SUBJECT('Tech Talk Long MessageDemo')

This is a test message.
A new line.

A new paragraph.
```

You can also send physical files, logical files, or source files as email with SNDDST. This is an example of SNDDST used to send a file:

```
SNDDST TYPE(*FILE) TOINTNET((murphyfa@uiuc.edu)) DSTD('Test Message') +
MSG(*NONE) DOCFILE(LIB/FILE) DOCMBR(*FIRST) DOCTYPE(*FFT) +
SUBJECT('Tech Talk File Demo')
```

This command places the contents of the file in the body of the email message. The trick is to specify a DOCTYPE of *FFT or 2, causing the file to be converted into ASCII text. There are limitations, however. SNDDST adds no carriage return or line feed. Line-feeds characters to the ends of records, so the beginning of the next record starts where the previous one stopped. In addition, source file records have their line numbers and date fields at the front, and packed decimal and binary fields are clobbered when SNDDST tries to convert their bytes into ASCII text.

Using the Copy to PC Document (CPYTOPCD) and Copy from PC Document (CPYFRMPCD) commands is a quick-and-dirty way to get the CR (Carriage Return) and LF (Line Feed) characters into a file (and get rid of the line numbers and date fields in source files). Translation Format (TRNFMT) (*TEXT) on CPYTOPCD adds the CR and LF, and TRNFMT (*NOTEXT) on CPYFRMPCD keeps them in the new file. The new file can have any record length, even 1 byte. SNDDST reads it as a stream of characters, and record length doesn't matter. That's what causes the formatting problem in the first place. When you send the new file with SNDDST, the text is formatted properly. Of course, if your file has a very long record length, it wraps around anyway because most email readers can't properly display something that wide. Here are some examples:

```
CPYTOPCD FROMFILE(LIB/OLDFILE) TOFLR(FOLDER) FROMMBR(MEMBER) +
TODOC(PCDOX.TXT) REPLACE(*YES) TRNTBL(*DFT) TRNFMT(*TEXT) *

CPYFRMPCD FROMFLR(FOLDER) TOFILE(LIB/NEWFILE) FROMDOC(PCDOC.TXT) +
TOMBR(MEMBER) MBROPT(*REPLACE) TRNTBL(*DFT) TRNFMT(*NOTEXT)

SNDDST TYPE(*FILE) TOINTNET((murphyfa@uiuc.edu)) +
DSTD('Test Message') MSG(*NONE) DOCFILE(LIB/NEWFILE) +
DOCMBR(*FIRST) DOCTYPE(*FFT) SUBJECT('Tech Talk File Demo')
```

An email attachment is a better way to send a file, and this is where the *DOC information type comes in. First, copy the file to a PC document just as you did previously. Then, send the PC document. SNDDST looks something like the following example:

```
SNDDST TYPE(*DOC) TOINTNET((murphy@uiuc.edu)) DSTD('PC Document') +
DOC(PCDOC.TXT) FLR(FOLDER) SUBJECT('Attached File')
```

You still have to worry about packed decimal and binary fields being mangled, but everything else comes through okay.

Before you can use SNDDST as just described, you must prepare your system. You must first have SNADS, TCP/IP, and Simple Mail Transfer Protocol (SMTP) all configured and working properly. All of that is a tad beyond the scope of this section. See the *OS/400 TCP/IP Configuration and Reference* manual if you haven't done all of that stuff already. Next, if you are not using V4R3, you need to apply one set of the following PTFs:

- SF45257 and SF48224 for V4R2.
- SF45226 and SF48223 for V4R1.
- SF45415 for V3R7.
- SF45328 for V3R2.

On all OS versions, the Change Distribution Attributes (CHGDSTA) command must be executed to make SNDDST work with the Internet, and, for CHGDSTA to work, you must make an entry for a gateway to SMTP in your system directory. Details are on the help screens for the CHGDSTA command.

—*Guy Murphy*

Set Up a Postmaster Account on the AS/400 with SMTP

Q: I've set up Simple Mail Transfer Protocol (SMTP) on my AS/400 so that my OfficeVision users can exchange email with users on the Internet. On most email systems, a profile is set up so that a remote user having a problem sending a message can ask a question to get the problem resolved. For example, a remote user who tries to send a message to an address that doesn't exist will often want to ask if the email address is different or if the person no longer works for the company. Usually, you'd send an email message to postmaster@yourdomainname.dom in that case. Is there a way to accomplish that with SMTP on the AS/400?

A: Yes, there is. Using the Add Directory Entry (ADDDIRE) command, you can create a postmaster account. Use a user ID of POST and an address of MASTER. Set the user ID to someone who isn't already set up in the distribution directory. That person will start getting messages for the postmaster user through SMTP.

—Jim Hoopes

FTP

Trimming Blanks with FTP

Some folks have problems when FTPing a file from the AS/400 to a PC: It tends to trim blanks from the end of the record. So, if you have a record that is 100 characters long and the longest record is only 85 characters long (followed by 15 blanks), the record length on your PC will be 85. To fix this, use the following command before sending the data to tell the AS/400 not to trim ending blanks:

```
LOCSITE TRIM 0
```

You can also use the following commands:

```
LOCSITE TRIM 1
LOCSITE TRIM
```

The former of these two commands will trim blanks; the latter will display the current value for TRIM.

—Bradley V. Stone

FTP and *SAVF

Can I Convert a PF to a *SAVF with FTP?

Q: I used FTP to transfer a save file from one AS/400 to another, but I forgot to create the save file on the receiving AS/400 first. Now, I have a physical file. Can I convert this physical file back to a save file?

—Dale Monti

A: You cannot take the physical file and create a save file from it, but you can re-FTP the file from your PC. Use the following commands to recreate your save file without losing the data in it. Assume, for example, that you inadvertently created a physical file named DATA1 on your AS/400. It should have been a save file, but you forgot to create the save file first. To transfer the data from DATA1 into a new save file in library QGPL, do the following:

1. From an AS/400 command line, create a save file named NEWSAVF in a library. (In this example, I'll use library QGPL):

```
CRTSAVF QGPL/NEWSAVF
```

2. Now, start the FTP on the AS/400 and use the following FTP commands to move the data from your physical file (that should have been a save file) into the new save file in QGPL:

```
ftp system name
user Your User ID
pass Your Password
binary
put DATA1 QGPL/NEWSAVF
quit
```

—Chuck Pence

Create a Save File Before You Transfer a Save File

Q: I am trying to transfer a save file (*SAVF) from one AS/400 to another using FTP. Everything seems to work fine until I try to restore the save file on the target system. I get a message CPF3782, "File {save file name} in {library name} not a save file." Can you help?

A: Before you perform the transfer, create a save file on the target system. You can create the save file on the target system in your FTP session with the FTP subcommand quote and the server command rcmd (as illustrated in Figure 20.4).

```
binary
 quote rcmd CRTSAVF TGTLIB/SAVEFILE
 put srclib/savefile tgtlib/savefile
 quote rcmd RSTOBJ OBJ(PGMX) SAVLIB(SRCLIB) DEV(*SAVF)
                     SAVF(TGTLIB/SAVEFILE) RSTLIB(TGTLIB)
```

Figure 20.4: Using FTP, this sample code allows you to transfer a save file.

When you transfer the file with the FTP put subcommand, be sure to specify the name of the save file you created on the target system as the remote file name. (The remote file name is the same as the local file name if the remote file name is not specified.) Figure 20.4 illustrates a sample FTP session that transfers a save file to a target system and restores the object contained in the save file to library TGTLIB.

—Richard Shaler

FTPing Save Files

Here's a very easy method of saving data to an AS/400 save file, and then emailing it to someone at a remote site to load and restore on their AS/400.

1. Save the object(s) or library(s) in an AS/400 save file (*SAVF) using the standard AS/400 Save Object (SAVOBJ) or Save Library (SAVLIB) commands.

2. Start an FTP session on your PC (or on your AS/400, depending on which one you are most comfortable using).

3. Here's the key to making this work correctly. Enter the FTP command binary in lowercase letters. This will cause the data to be transferred as a binary image.

4. Now use the FTP put command or get command (the command you use will depend on whether you are sending data to the PC from your AS/400 or getting data from your AS/400 and loading it onto the PC), to load the save file onto your PC.

5. Email the file to wherever you want it to go.

6. At the remote site, use the same steps as outlined above to put the save file on the remote AS/400. Make sure the remote site also uses the FTP binary command to ensure the file is transferred as a binary image.

7. Use the AS/400 command Restore Object (RSTOBJ) or Restore Library (RSTLIB) to restore the save file.

That's all there is to it!

—Shannon O'Donnell
—Brad Stone

Transferring AS/400 Save Files with FTP

AS/400 save files are the PKZIP of the midrange. I transfer them all the time using FTP. I can quickly transfer a save file from one AS/400 to another, or I can download a save file to my PC and send it to whomever as an email include. The process is simple, and, if you are comfortable with FTP, I can summarize it in one word: binary. The AS/400's implementation of FTP assumes, unless told otherwise, that file transfers are for text files and, as such, will require EBCDIC-to-ASCII translation. Just think what that'll do to your save file! The binary FTP directive says, "don't do that" to the AS/400.

In a nutshell, then, here's how to FTP a save file from an AS/400 to a PC and from an AS/400 to another AS/400.

Downloading an AS/400 Save File to Your PC

1. Start FTP on your PC (FTP.EXE).

2. Type in the open command, and FTP will prompt you for the TCP/IP address or Host Name of the AS/400 system from which you are transferring files. It will also prompt you for the user profile name and password to use when logging on to your AS/400. Enter the appropriate information.

3. Enter the binary directive (or the abbreviation, BIN).

4. Get your save file to the PC with the get Library/SaveFileName PCFileName command.

For instance, to download a save file called SAVESTUFF in my DENONCOURT library to the C:\work\mc\september directory on my PC, I entered the following command:

```
get DENONCOURT/SAVESTUFF c:\work\mc\september\SaveStuff.sav
```

Uploading a PC File That Contains an AS/400 Save File to an AS/400

1. Create or clear the AS/400 save file that is to be uploaded. If the save file does not already exist, FTP will create a file for you, and it will not be of type *SAVF:

```
quote rcmd CRTSAVF DENONCOURT/RESTORSTUF
```

Or

```
quote rcmd CLRSAVF DENONCOURT/RESTORSTUF
```

2. Start FTP on your PC (FTP.EXE).

3. Type in the open command, and FTP will prompt you for the necessary login information, including TCP/IP address or Host Name of the AS/400 system, user profile name, and password.

4. Enter the binary directive (or the abbreviation, BIN).

5. Transfer the file from your PC to the AS/400 with FTP's put command:

```
put c:\work\mc\september\SaveStuff.sav DENONCOURT/RESTORSTUF
```

Transferring an AS/400 Save File Between AS/400s

1. Start FTP:

```
STRTCPFTP RMTSYS(Other400DomainName)
```

2. Enter your user profile and password.

3. Create or clear the AS/400 save file that is to be uploaded. If the save file does not already exist, FTP will create a file for you, and it will not be of type *SAVF:

   ```
   quote CRTSAVF DENONCOURT/RESTORSTUF
   ```

   Or

   ```
   quote CLRSAVF DENONCOURT/RESTORSTUF
   ```

4. Enter the binary directive (or the abbreviation, BIN).

5. Transfer the file from the local or current AS/400 to the host AS/400 with FTP's put command:

   ```
   put DENONCOURT/RESTORSTUF HOSTLIB/RESTORSTUF
   ```

   Optionally, you could transfer the file from the host AS/400 to the current or local AS/400 with FTP's get command:

   ```
   get HOSTLIB/RESTORSTUF DENONCOURT/RESTORSTUF (REPLACE
   ```

For the get command only, you need to use the (REPLACE parameter to clear out the save file on your AS/400 (the put command clears out the save file for you). Note that there is no closing parenthesis on this parameter.

Caution: Take care when transferring a save file from one AS/400 to another or you might overlay an important save file. It's easy to get confused if the library and file names are the same. FTP is so easy to use that you can easily update the new version of a library with an old version—effectively wiping out all of your updates. Just remember that the TCP/IP address that FTP uses is considered the host site. The FTP get command copies the file from the host to the local machine (the one from which you entered the Start TCP/IP FTP (STRTCPFTP) command). The FTP put command copies the file from the local machine to the host machine.

—Don Denoncourt

FTP and Commands

Execute Remote Commands with FTP

Q: Can you recommend a product on the AS/400 that will accept remote commands? I use the remsh (Remote Shell) command from our RS/6000 to execute processing on other UNIX platforms. We also have an RS/6000 communicating with our AS/400 using TCP/IP. I would like to execute CL commands on the AS/400 from the RS/6000.

A: You can execute AS/400 commands from a remote system by using FTP. After you FTP to your AS/400, use the QUOTE subcommand with the RCMD parameter followed by the command you want to execute. For example, if you want to run the Work with Active Jobs (WRKACTJOB) command, you would enter this from within an FTP session:

```
QUOTE RCMD WRKACTJOB OUTPUT(*PRINT)
```

—Robin Klima

FTP and IFS

Pick the Right Name Format

If you want to take advantage of the FTP support for shared folders, which IBM added to FTP in V3R1, you have to use the new FTP subcommand NAMEFMT. It allows you to change the default naming format of **Library/File.Member (NAMEFMT 0)** to **/QSYS.LIB/Library.Lib/Filename.File/Member.MBR (NAMEFMT 1)**.

While this might seem like more work to access regular system objects, it is the only way to access shared folders with FTP. For example, the files for Client Access for DOS are in directory /QDLS/QIWSFL2. But there’s a problem with the NAMEFMT command: It only works when an AS/400 is the FTP client. If you FTP from your PC to the AS/400, executing subcommand NAMEFMT produces an error. This is because the PC’s FTP client doesn’t know anything about name formats. You need to change the FTP server on the AS/400, which you can do with this command:

```
QUOTE SITE NAMEFMT 1
```

The FTP server on the AS/400 (not on the client) executes this command. The AS/400 will then understand the new naming format.

—Martin Pluth