# The Operating System Handbook

## or, Fake Your Way Through Minis and Mainframes

by Bob DuCharme

## OS/400

# Table of Contents

## This Section, the Rest of the Book

This is one part of the book "Operating Systems Handbook (or, Fake Your Way Through Minis and Mainframes)," which was originally published by McGraw-Hill as a $49.60 hardcover. Once they reverted the rights to me after it went out of print, I converted the original XyWrite files to DocBook XML and then used Norm Walsh's stylesheets (see www.nwalsh.com) and the Apache FOP program (see xml.apache.org) to convert that to Acrobat files. The six parts of the book are all available at http://www.snee.com/bob/opsys.html:

- Part 1: Introduction. Note that this part includes a new section explaining why I didn't update the book. I strongly suggest that, no matter how much or how little of the book you can use, you glance through the whole Introduction as well. The "Comments and Suggestions" part is now obsolete; my home page is now http:/www.snee.com/bob and my e-mail address is bob@snee.com.
- Part 2: UNIX. Everything described here should apply to Linux and its relatives.
- Part 3: OpenVMS. Basically, VMS. DEC was calling it "OpenVMS" at the time.
- Part 4: OS/400. The operating system for IBM's AS/400.
- Part 5: VM/CMS. An IBM mainframe operating system.
- Part 6: MVS. Another IBM mainframe operating system.

See www.snee.com/bob for information on books I've written since. In reverse chronological order, they are:

- *XSLT Quickly* is a tutorial and users guide to XSLT designed to get you writing stylesheets as quickly as possible.
- *XML: The Annotated Specification* is a copy of the official W3C XML specification with examples, terminology, and explanations of any SGML and computer science concepts necessary for a complete understanding of the XML spec.
- *SGML CD* is a users guide to free SGML software, most of which can be used with XML as well. The chapter on Emacs and PSGML, which requires no previous knowledge of Emacs, is available on the web site in English, Russian, and Polish.

One more thing: either I couldn't figure out the XSL `keep-together` property or version 0.18.1 of FOP doesn't implement it yet. Either way I apologize that some screen shots get split across page breaks.

# Chapter 12 OS/400: An Introduction

OS/400 is the operating system used on IBM's line of AS/400 minicomputers. "OS" stands for "Operating System," as it does in "OS/2," and "AS" stands for "Application System." No one uses any other operating system on the AS/400, and no one installs OS/400 onto any other computers. Because they're always used together, it's common to use the terms "OS/400" and "AS/400" interchangeably when we talk about using the computer.

The combination of OS/400 and the AS/400 has an odd mix of advanced and old-fashioned approaches to computing. On the one hand, the object-oriented approach of treating system resources and their interaction as objects and messages exists in few other operating systems used on a large scale in the business world (none of the ones covered by this book) but will be seen more and more in the coming years. On the other hand, certain aspects of OS/400 and the AS/400 show their roots in aging technology, such as IBM's encouragement of developers to use RPG and COBOL and the inclusion of an eight inch or five and a quarter inch disk drive without an option for three and a half inch disk drives. Because none of these links to older IBM technology are inherent parts of an AS/400 system, I'm sure that they will be brought up to date soon if they haven't been by the time you read this. The Integrated Language Environment (ILE), available with Version 2 Release 3 of OS/400, already promises to make things easier for OS/400 developers who use the C programming language.

While features like communications, built-in database support, transaction processing, and system security were added on to other operating systems over the years, these features were all part of the design of OS/400 from the start. This may be its greatest advantage, from a design point of view—IBM included these features as intrinsic parts of the operating system from the beginning instead of patching them in over time.

From a user's perspective—or even a system administrator's—the biggest advantage of OS/400 is its ease of use. While many operating systems offer menus to make things easier for the beginner, OS/400 offers access to a greater percentage of its capabilities through menus than any other popular operating system. Its command language isn't too bad either; while the abbreviations that form the commands give them a terse, strange appearance, they're actually pretty easy to figure out once you learn the logic behind the abbreviation system.

For the system administrator, the ease of use starts as soon as the AS/400 comes out of the box. IBM went to great trouble to make the AS/400 a "plug and play" computer, doing much of the setup that a customer needs before shipping it.

All this combines to make the AS/400 a very successful computer. It has been the silver lining on some of IBM's darkest clouds; in February of 1993, just when the public and stock market were taking the dimmest view of IBM's future as a major player in the computing world, IBM took out a full page ad in the New York Times that proclaimed in huge letters "You ain't seen nothing yet. The IBM AS/400. Success isn't complicated." The ad's copy bragged that IBM had "shipped over 200,000 AS/400s in just over four years—more than our closest competitor has in seven—to companies of all sizes...with a customer satisfaction rate just shy of 98%, we've got nothing to be shy about."

## 12.1 History

The 1960's saw the birth and growth of the minicomputer market. IBM got into the game fairly late; in 1969, eight years after DEC introduced the PDP-1, IBM introduced its first minicomputer: the System/3. Although IBM had the resources to provide better support than other minicomputer makers, it didn't have a better minicomputer. IBM also managed to charge more than their competitors, so the System/3 provided no threat to DEC's market and became a historical footnote.

In 1975, IBM announced the System/32. This batch oriented, single-tasking computer was the beginning of what people now refer to as the "System/3X" family of computers. Neither the System/32 nor its multi-user, multi-tasking 1977 successor, the System/34, took a firm hold in the marketplace, but the 1978 System/38 and the 1983 System/36 became very successful.

Users found the System/36 easy to use, and IBM kept this in mind when designing its replacement. The popularity of the simple interface on the System/36 and the large installed base of System/36 and System/38 minicomputers were key factors in the design of the AS/400, which IBM introduced in 1988. While the AS/400 and OS/400 had an entirely new architecture, subsystems were built in to ensure that the more than 8,000 existing System/3X application packages could run on it. (You can run System/38 programs in the AS/400's System/38 mode without modification, but you must recompile System/36 programs before you can run them in the AS/400's System/36 mode.) This gave the AS/400 a big start in available applications—always an important issue with a new computer and operating system.

IBM currently offers five different series of AS/400s: the B, C, D, E, and F series. Within each series, a number attached to the letter identifies how that model relates to others in that series. The higher the number, the greater the power. For example, the F series, when first introduced, offered the F02 as the simplest model and the F95 as the most powerful.

You can upgrade many models to more advanced AS/400s. For example, models from the B and C series can be upgraded to the D series. This upgradability has always been a big selling point for the AS/400; if you get one for your small company and your company grows, your AS/400 can grow with you.

IBM had an amusing way of demonstrating this to the public. You probably remember how IBM's original television ads for the PS/2 featured all the major characters from the television show MASH except for Hawkeye. They saved him for the AS/400 ads. (As far as I know, these were not run on television, but run as print ads aimed at a more specialized audience.) To symbolize the ability of the AS/400 to grow with your business, he was shown watering a little plant next to a small AS/400; in the next picture, he stood proudly by a much bigger plant—ostensibly, the same as the one in the first picture, but grown larger—next to a much bigger AS/400. Warm and friendly as Alan Alda, but capable of providing all the power you need when you need it—that's the AS/400.

**BUZZWORD** *Blue shop* This is actually two buzzwords: "blue" means IBM in the computer world the same way "yellow" means "Kodak" in the world of photography. It's an adjective meaning "of or pertaining to IBM." As a noun, people often say "big blue"; for example, a trade journal might have the headline "Big Blue Announces New AS/400 Models."

"Shop" refers to a given company's collection of minicomputers and mainframes. A "blue shop" means that a particular company uses only IBM's minis and mainframes.

### 12.1.1 Today

As more and more AS/400's are installed, and more and more applications are written for it, backwards compatibility with the System/3X computers has become less of an issue. As PCs and networked PC applications become more prevalent and more powerful, the AS/400's relationship to this growing segment of the computer world, and not its relationship to IBM's earlier minis, has come to define Big Blue's positioning of the AS/400's advantages.

Three features play a key role in this new positioning:

- *Database management.* The AS/400 was designed from the start to be a database management machine. It has a relational database manager built in as an integral part of both the operating system and the machine itself. You don't buy it as a separate piece of software; in fact, if you do install another database manager on the AS/400, it must translate everything into the AS/400 database manager's terms in order to perform any work. To learn more about the AS/400's database management capabilities, use the search index described in section 13.4, "Available On-line Help," to search for information about DFU (Data File Utility) and IDDU (Interactive Data Definition Utility).

- *Communications.* The AS/400 communicates with other computers—mainframes, PCs, other AS/400s, and non-IBM minis—better than any past or present IBM computer. It can communicate with other computers using protocols such as Ethernet, SNA, OSI, ISDN, TCP/IP, Novell NetWare, and others. It can support connections to three different Local Area Networks at once.

- *Security*. Features for the implementation and maintenance of data security are also built right into the operating system and the machine itself. You can define data security in terms of objects, users, workstations, files, records, and fields.

These three features combine to make the AS/400 an ideal candidate for a database server. IBM had ideas about this from the beginning, but in the original grand plan the desktop computers attached to these servers were supposed to be PS/2s running OS/2. Over time, IBM realized that there was a greater advantage in making the AS/400 capable of being a database server for any other computer that needed it. While the database and security features mentioned above were built into the AS/400 from the beginning, many of the communications protocols were added over time (particularly with the introduction of the D series in 1991) as IBM began to reposition the AS/400 as a more general-purpose database server. Version 2 of the OS/400 operating system made it easier for developers to write applications that store and use PC files on an AS/400; this encouraged developers to write client/server applications that use the AS/400 as the server.

IBM also offers the PC Support package for the AS/400, which links PCs to an AS/400. PC Support makes it easier for PCs to act as full-featured AS/400 terminals and to use disks and printers attached to an AS/400 system.

The AS/400 doesn't always have to be a server in a client/server relationship; it can also be the client, requesting data from another system. Its extensive peer-to-peer communications abilities give it a great deal of flexibility, allowing the same AS/400 to be the server for one application and the client for another.

## IBM's Minicomputer: The AS/400 or the RS/6000?

Shortly after introducing the AS/400, IBM introduced the RISC System, or RS series of computers. The RS computer is essentially an engineering workstation that runs AIX, IBM's version of UNIX. It can be run as a single- or multiple-user machine.

It sounds like a minicomputer. How does it differ from the AS/400, the successor to the System/3X line and therefore IBM's current main entry in the minicomputer market? The RS/6000's various technical differences from the AS/400 reflect one key point: it's aimed at a different audience. A scientist or engineer who must write his own C or C++ program to attack a complicated math problem that produces complex graphical output would not use the AS/400, a multiuser machine optimized to run common database applications such as inventories and payrolls. The technical user wants a machine that he or she can dedicate to the problem at hand without being slowed down by other users' programs. The RS/6000 user is familiar with UNIX and UNIX tools, knows how to combine them to solve technical problems, and wants to leverage this knowledge.

The RS machines and the AIX operating system are designed for this user. While the AS/400 competes in the marketplace with business machines like the VAX and Hewlett-Packard's minicomputers, the RS/6000 competes with other engineering workstations from companies like Sun and Silicon Graphics. And it competes very well; it has a reputation as a well-built, powerful, reasonably priced engineering workstation. This made the RS/6000, like the AS/400, a bright spot in IBM's dark days of the early 1990s.

In fact, the RS/6000 has done so well that it has ended up competing with the AS/400 after all. One of the AS/400's most promising roles is that of a PC file server, and some find the RS/6000 more suited to this than the AS/400—regardless of the applications being run on the PC.

For information on using the RS/6000 computers, see this book's chapters on UNIX, particularly the sidebars in Chapter 2 titled "ULTRIX? XENIX? AIX? AUX? POSIX? DYNIX? MACH? SunOS?" and "Workstations."

#### 12.1.1.1 Popular OS/400 Software

As mentioned earlier, OS/400's built-in database manager is one of the main reasons people use the system. Another popular package available to run under OS/400 is OfficeVision. OfficeVision offers many general-purpose features needed by almost any office: calendar and scheduling software, word processing, electronic mail, and integration of these facilities with the AS/400's built-in database. Although message sending and the SEU text editor are built into OS/400, the OfficeVision alternatives are more powerful and easier to use. (SEU has many features that still make it better for entering program source code, since OfficeVision's word processor is geared more toward correspondence and document composition.)

# Chapter 13 Getting Started with OS/400

## 13.1 Starting Up

The first thing you see when you connect to an AS/400 is its sign-on screen. Figure 13.1 provides an example.

```
                            O'Rourke Enterprises
                                 AS/400-F45
                                    V2R2


  System  . . . . :  NEPAS4
  Subsystem . . . :  QINTER
  Display . . . . :  DISP07


  User  . . . . . . . . . . . . . .
  Password  . . . . . . . . . . . .
  Program/procedure . . . . . . . .
  Menu  . . . . . . . . . . . . . .
  Current library . . . . . . . . .


                                        (C) COPYRIGHT IBM CORP. 1980, 1992.
```

**Figure 13.1 AS/400 sign-on screen.**

Your screen may not look exactly like Figure 13.1. It may have additional information, and it may not tell you the release of OS/400 being used (in the example, "V2R2" means "Version 2 Release 2"). It will tell you three things about where you are signing on:

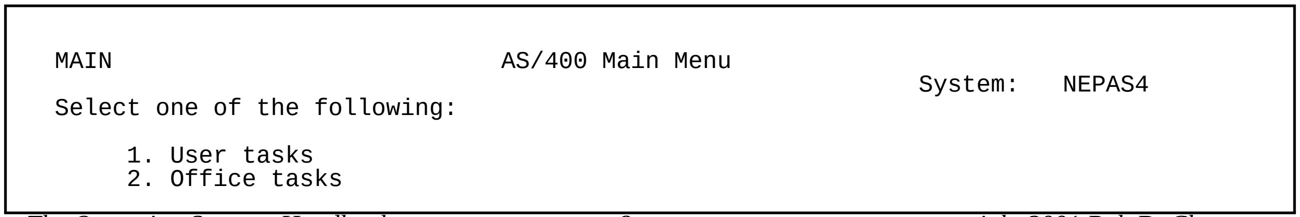
| | |
|---|---|
| `System` | The name assigned by the system administrator to the particular AS/400 on which you have your account. |
| `Subsystem` | The system administrator may divide various aspects of the system into different areas known as "subsystems." Don't worry about it. |
| `Display` | The name assigned to the terminal you are using to sign on. |

Below this information, several fields appear for you to complete. You only need to fill out the first two. Use your Field Exit key (when emulating a terminal, this will probably be the Tab key) to move your cursor forward from field to field. If you make a typing mistake, use your Backspace key to reposition your cursor and fix the mistake.

| | |
|---|---|
| `User` | The user ID that represents your identity on the system. If you don't have one, you can't sign on, so contact your system administrator. It doesn't matter whether you type in your ID in upper or lower case; it displays in all capital letters.<br><br>If you enter a user ID that the system doesn't recognize, it will tell you. For example, if Joe User mistypes his JOEUSER user ID as JEOUSER, OS/400 displays a message at the bottom of the sign-on screen telling him "User JEOUSER does not exist." |
| `Password` | The password that goes with your ID. If you enter the wrong password for the entered user ID, the system tells you "Password not correct for user profile." The user profile is the collection of information stored about a particular user—the user ID, the password, and the access rights that user has to the various parts of the system. |
| `Program/procedure` | If you know the program that you want to run as soon as you are signed on, you can enter its name here. |
| `Menu` | All menus in OS/400 have names. If you want to go directly to a menu other than the currently designated initial menu, enter its name here. |
| `Current library` | A library is a collection of objects, much like a subdirectory on other operating systems is a collection of files. The current library is the first place the system looks when you request the use of a particular object. If you want a library other than the default one to be the current library for this session, enter its name here. For more information on libraries, see section 13.3, "How Files Are Organized." |

After you have signed on successfully, OS/400 displays the initial menu.

This will probably be some variation of the AS/400 Main Menu (Figure 13.2), but your system administrator may set another menu as your initial one. If there is a specific menu that you prefer to designate as the initial menu, you can easily set this up; see section 16.2.1, "The Automatic Signon Command File," for more information.



```
MAIN                           AS/400 Main Menu
                                                            System:   NEPAS4
Select one of the following:

     1. User tasks
     2. Office tasks
```

```
      3. General System Tasks
      4. Files, libraries, and folders
      5. Programming
      6. Communications
      7. Define or change the system
      8. Problem handling
      9. Display a menu
     10. Information Assistant options
     11. PC Support tasks

     90. Sign off

Selection or command
===>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=User support
F23=Set initial menu
(C) COPYRIGHT IBM CORP. 1980, 1992.
```

**Figure 13.2 OS/400 main menu.**

## 13.1.1 Finishing Your OS/400 Session

As you will see in section 13.1.2, "Entering Commands," there are two ways to do just about anything in OS/400: by entering a command or by picking a choice off of a menu. Menu choices are numbered, and from the OS/400 main menu (Figure 13.2) signing off is always choice number 90.

Picking a command off of a menu is actually similar to entering a command—you're just entering a number instead of typing out a word or phrase. To sign off, type 90 at the `===>` prompt.

To sign off by entering a command, type `SIGNOFF`. Although menu choice 90 shows "sign off" as two words, remember to enter the command as one word.

The command has one particular advantage over using the number to designate the menu choice: this menu choice is only available on the main menu, while you can type `SIGNOFF` anywhere that you see the `===>` prompt.

## 13.1.2 Entering Commands

Along with `SIGNOFF`, any OS/400 command can be entered anytime you see the `===>` prompt.

We call these commands "CL commands" because they make up the OS/400 Command Language. (When I say "command prompt," I'm referring to the `===>` prompt.) As you'll see in section 16.2, "Command Files," OS/400 command files are known as "CL programs."

Most menus have the phrase "Selection or command" above this prompt. This means "type the number of the menu choice you want to make or a command at this command line." Some menus have no `===>` prompt, but your cursor still appears at the bottom of the screen, waiting for you to type something. If the line above your cursor says "Type a menu option below," then you can't enter a command; you must enter a menu choice number. You might not even notice these clues indicating that you must enter a number and not a command, but if you start entering a command and your cursor jumps back to the first character that you typed as soon as you enter the second character, don't panic; you've found one of these menus.

Much AS/400 literature talks about how intuitive and English-like the CL commands are. When you see command names like `SNDMSG, DLTF, STRSEU` and `DSPMSG`, you're bound to wonder: if this is English-like, then I guess UNIX commands aren't so bad after all!

Actually, CL commands aren't too bad once you learn the abbreviation system. The words of a command are usually abbreviated to three (or fewer) letters and then strung together into verb-object or verb-modifier-object order. The verb part is always three letters. (There are a few commands that deviate from this system, like `GO`, which you use to jump directly to a specific menu, and `SIGNOFF`.)

Keeping this system in mind, `SNDMSG` doesn't look as cryptic anymore; it means "Send Message." `DLTF` is "Delete File." Once you know that the Source Entry Utility (SEU) is the name of the OS/400 text editor, it's not too hard to remember that `STRSEU` means "Start Source Entry Utility."

Many commands use STR as an abbreviation of "start." DSP, which means "display," is also popular; `DSPMSG`, or "display message" is only one of many commands that begin with these three letters. Once you get used to the abbreviations for common verbs like STR and DSP and common verb objects like MSG, you can figure out many commands on your own.

The following shows some other common verb abbreviations:

| | |
|---|---|
| `CRT` | create |
| `WRK` | work with |
| `CHG` | change |

Certain noun abbreviations also come up often:

| | |
|---|---|
| `LIB` | libraries |

| | |
|---|---|
| `OBJ` | objects |
| `CMD` | commands |

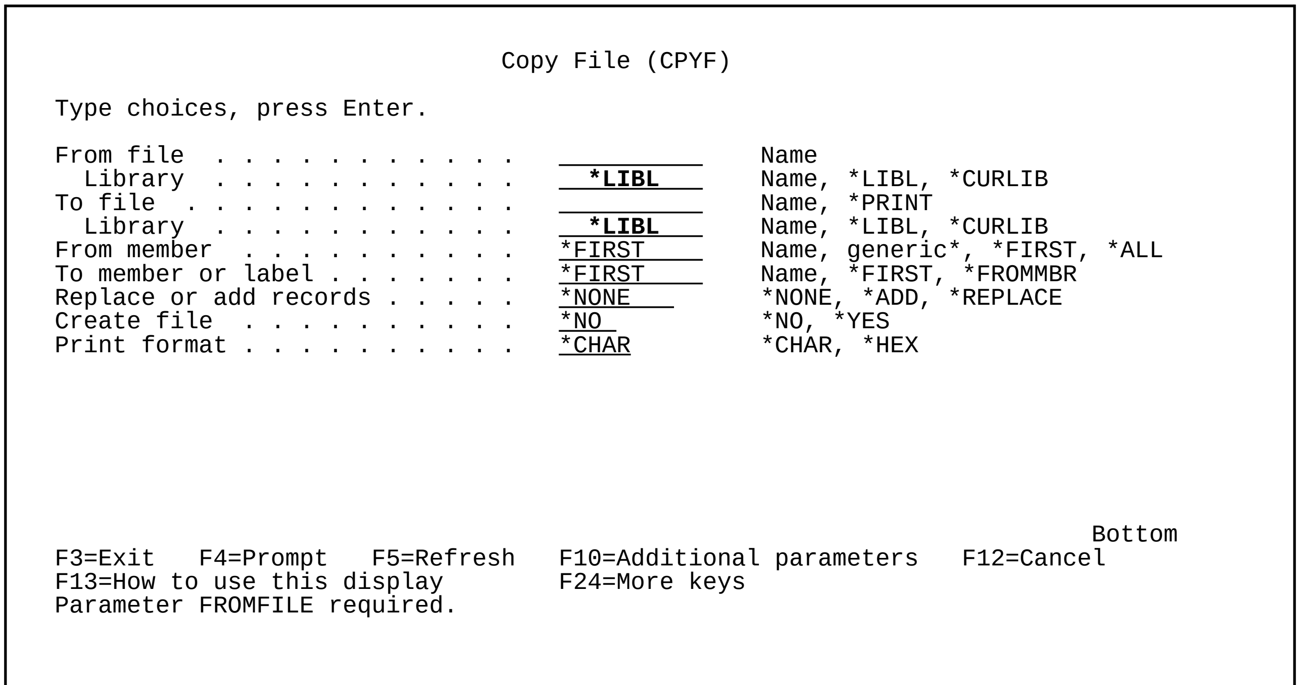
#### 13.1.2.1 Command Parameters

When a command needs parameters, you enter each one in parentheses preceded by the name of the parameter. For example, the following command renames a file object called `OLDNAME` with a name of `NEWNAME`:

```
RNMOBJ OBJ(OLDNAME) OBJTYPE(*FILE) NEWOBJ(NEWNAME)
```

If you can't remember the name or order of the parameters, there is plenty of help.

The easiest way to deal with the parameters is the prompt key, F4. If you enter a command name and press F4 instead of Enter, OS/400 displays the *command prompt display*, which displays the parameters for that command on a form for you to fill out. To make things even easier, it highlights the required parameters that don't have default values and displays a message at the bottom telling you the next required parameter to fill out. Figure 13.3 shows the command prompt display that you get if you press F4 immediately after entering `CPYF` at the command prompt.



```
                               Copy File (CPYF)

 Type choices, press Enter.

 From file  . . . . . . . . . . .   __________    Name
   Library  . . . . . . . . . . .     *LIBL       Name, *LIBL, *CURLIB
 To file  . . . . . . . . . . . .   __________    Name, *PRINT
   Library  . . . . . . . . . . .     *LIBL       Name, *LIBL, *CURLIB
 From member  . . . . . . . . . .   *FIRST        Name, generic*, *FIRST, *ALL
 To member or label . . . . . . .   *FIRST        Name, *FIRST, *FROMMBR
 Replace or add records . . . . .   *NONE         *NONE, *ADD, *REPLACE
 Create file  . . . . . . . . . .   *NO           *NO, *YES
 Print format . . . . . . . . . .   *CHAR         *CHAR, *HEX







                                                                       Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
 F13=How to use this display        F24=More keys
 Parameter FROMFILE required.
```

**Figure 13.3 Command prompt display for the Copy File command.**

Another way to bring up the command prompt display is to precede the command with a question mark (for example, `? CPYF`). This is a nice trick for making your CL programs a little more interactive—instead of filling in a command's parameters in the CL program, so that the command is executed with the same parameters every time someone runs the program, a question mark before the command brings up that command's command prompt display. This allows the CL program's user to set different parameters each time the CL program is run.

As with the signon screen, you use the Tab key to move from field to field. Press Enter when you're ready to execute the command or F3 to exit out of the command prompt display. If you press Enter before you've filled out all the required fields, your cursor jumps to the first mandatory field that you neglected to fill out and a message appears at the bottom of the screen telling you that parameter's name and the fact that it is required.

If you press F4 after filling out any of a command's parameters at the command line, those parameters will be filled in for you on the command prompt display.

Certain commands don't require any parameters, but won't do anything unless you include some. For example, the Change Profile command (`CHGPRF`) changes various settings that control your signon ID's working environment. If Joe User enters `CHGPRF` with no parameters and presses Enter, he'll see the message "User profile JOEUSER changed." Nothing about it was really changed, though, because he didn't indicate any changes to make. Although it has no mandatory parameters, the Change Profile command has so many optional ones that AS/400 users nearly always use F4 with it instead of Enter; this allows them to see the names of the various values that can be reset.

After entering a command name and any of its parameters at the command line, you can also press F1 to display the on-line help about that command. See section 13.4, "Available On-line Help," for more information.

#### 13.1.2.2 Positional Parameters

When entering a command on the command line, some parameters require you to only enter their value, without the parameter name or parentheses. These are called "positional parameters" because each one's position on the command line tells OS/400 which parameter it is. The use of positional parameters is most common for required parameters, since it saves you some typing.

A common positional parameter is `FILE()`, which is often the first parameter for many commands. For example, to delete a file in the current library called `BUDGET95`, you might enter this,

```
DLTF FILE(BUDGET95)
```

but it's easier to enter this:

```
DLTF BUDGET95
```

It's not unusual to leave out the parameter names and parentheses for some of a particular command's positional parameters and to include them for other optional ones, all on the same command line.

For a more detailed case study of the use of positional parameters, see section 14.1.4, "Copying Files."

### 13.1.2.3 Case Sensitivity

OS/400 doesn't care whether you enter commands, parameters, or parameter names in upper or lower case. To delete a file called `DRAFT1` in the library `BUDGET94` (See section 13.3, "How Files Are Organized," for more on libraries), you could type any of the following three commands:

```
 DLTF FILE(DRAFT1)
 dltf file(draft1)
 DlTf FiLe(DrAfT1)
```

### 13.1.2.4 The Four Types of Displays

OS/400 has four different kinds of screens. IBM terminology calls them four different kinds of "displays." All four display types have two things in common: a title at the top to let you know exactly which display it is and a list of available function keys across the bottom of the display. As section 13.1.2.5 ("Important Special Keys") shows you, the function keys always offer you a way to back out of any screen that you displayed accidentally.

The main screen that you see after signing on (Figure 13.2) is an example of a *menu display*. Menu displays include a command prompt where you can type either CL commands or the number of the menu choice that you want to select. The upper-left corner shows the name of the menu.

Knowing the names of the popular menus is very useful, because entering the `GO` command at any command prompt lets you jump directly to any menu. For example, to jump to the `MAIN` menu, enter

```
GO MAIN
```

at the command prompt.

An *entry display* is a form for you to fill out. Section 13.1.2.1, "Command Parameters," described how you can press F4 instead of entering a command's parameters if you are unsure of those parameters. The command prompt display that this brings up (Figure 13.3) is a good example of an entry display.

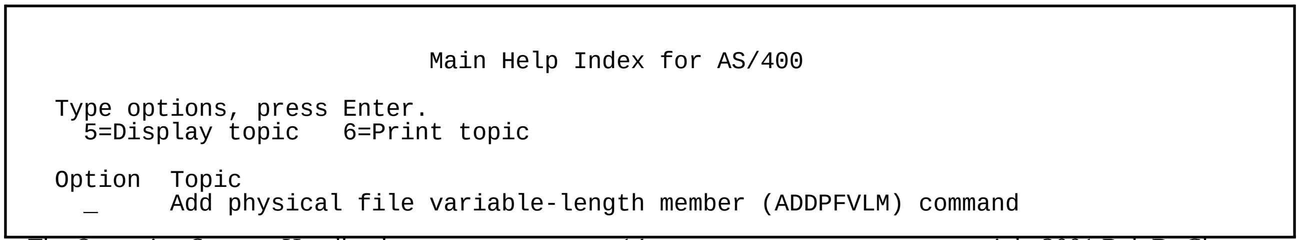
Sometimes there are hints to the right of each field on an entry display about what you can enter in that field. There are three types of clues:

| | |
|---|---|
| `Name` | A name that you either make up or should already know. For example, when you copy a file, you should know its name and you will make up a name for the copy. In Figure 13.3, `Name` is an option for the first six fields. |
| `generic*` | A name with the asterisk wildcard that applies to multiple names. If you are issuing a command that will display a hundred object names, and entering `JULYBUD` will list only the `JULYBUD` object, then entering `JULY*` will list the names that begin with the letters "JULY." For more information, see section 13.2.1, "Wildcards." |
| `*PREDEFVAL` | A *predefined value* is a name that has a specific, defined meaning to OS/400. These are usually spelled in all capital letters and begin with an asterisk. For example, when you send a message to another user with the `SNDMSG` command, you enter the message recipient's ID in the "To user profile" field. On the right of the blank where you enter the recipient's name, it offers the choices "Name, *SYSOPR, *ALLACT." The `Name` part means that you can just type in someone's user ID. `*SYSOPR` and `*ALLACT` are predefined values; `*SYSOPR` represents the system operator, so if you have a problem with the system and want to send a message to the system operator you don't need to know which system operator is on duty and his or her user ID. `*ALLACT` means "all active users." It lets you send a message to everyone who is currently signed on. (This option would be useful to the system operator to warn everyone of system problems or impending down time.) |

An entry display often displays more fields than you need to fill out. To make it easier for you to find the mandatory fields, they are highlighted for you.

Some of an entry display's fields may be filled out for you with default values. To change them, move your cursor there with your Tab or cursor keys and type a new value over the displayed one.

When you ask OS/400 to list information (for example, the files in a library), it often displays the information on a *list display*, which allows you to perform actions on items in the list. Figure 13.4 shows an example of a list that is displayed when you ask the on-line help about help topics that cover "programming."



```
                             Main Help Index for AS/400

  Type options, press Enter.
    5=Display topic   6=Print topic

  Option  Topic
    _     Add physical file variable-length member (ADDPFVLM) command
    _      Add program (ADDPGM) command
    _      Analyze program (ANZPGM) command
    _      APPC
    _      Application program
    _      Apply program temporary fix (APYPTF) command
    _      APPN
    _      Attention-key-handling program
    _      Automatic link and external reference
    _      BASIC
    _      Branch instruction
    _      Breakpoint program

                                                                   More...
  Or to search again, type new words and press Enter.
    programming

  F3=Exit help   F5=All topics   F12=Cancel   F13=User support
```

**Figure 13.4 Entry display of search help topics for "programming."**

The first column of a list display is the `Option` or `Opt` column. It shows a blank next to each list item where you enter a number to indicate what you want to do with that list item. Above the list, a key to the actions shows the numbers you can enter in the options column and what they do. For example, entering the number 6 next to a help topic title in Figure 13.4 tells OS/400 to send a copy of that help topic to the printer. Use your Tab or cursor keys to move your cursor to the appropriate place in the `Opt` column. (For more information on help topic lists, see section 13.4.1, "The Search Index.")

List displays sometimes have a command line at the bottom. In addition to regular CL commands, you can type any additional information that OS/400 needs to perform an action indicated by a number in the option column. If this is the case, the list display will let you know just above the command line. For example, if the key to the actions above the list shows that you can enter 1, 2, 3, 4, or 5 in the options column and entering 2, 3, or 5 requires additional information, you will see the following message just above the command line:

```
Parameters for options 2, 3, 5, or command
```

In other words, you can enter the parameters for options 2, 3, or 5 at the command line or a regular CL command.

A list or menu display may be too long to fit on your screen. If it fits, the lower-right of the screen will say "Bottom"; if it's too long, the lower-right will say "More..." (as in Figure 13.4). If there is more, the Page Down and Page Up keys let you scroll through the list. (See section

13.1.2.5, "Important Special Keys," for more on the use of these keys.)

## 13.1.2.5 Important Special Keys

We've already seen three special keys: F4 displays an entry screen that prompts you for a command's parameters, and Page Up and Page Down scroll through a display that doesn't fit on your screen.

When function keys are available to help you, the bottom of the display lists as many function key descriptions as will fit. Certain function keys nearly always have the same meaning:

| | |
|---|---|
| `F1` | *On-line Help* displays information about the current display and your options for what to do with it. See section 13.4, "Available On-line Help," for more information. |
| `F3` | *Exit* ends the current program (for example, on-line help or a command prompt display) and returns to the screen that was displayed when you called that program. (Compare this with F12.) |
| `F4` | *Prompt* displays the command prompt display, an entry screen whose fields prompt you for the parameters of the command at the command line. |
| `F9` | *Retrieve* retrieves the previously entered command to the command line. This can be particularly useful to correct a typo in a long, complicated command that didn't execute because of the typo. Just press F9, correct the mistake, and press Enter.<br><br>F9 can also teach you the parameters that a command needs; if you press F4 and use the command prompt display to enter a command's parameters, pressing F9 later retrieves the command to the command line as if you had typed it out without the aid of the command prompt display.<br><br>Retrieve does not retrieve menu choice numbers typed at the command line, but only actual commands. Pressing it repeatedly retrieves earlier commands from your OS/400 session. |
| `F12` | *Cancel* leaves the current display and returns to the one before it. This is useful when you are searching through help screens and you want to return to one you just saw without leaving the help program.<br><br>If you're looking at the first screen in a particular program, this key acts like F3, returning you to the screen that was displayed when you |

| | |
|---|---|
| | called that program. |
| `F13` | *User Support* displays the User Support and Education menu. This offers a menu-driven way to access the various kinds of on-line help available for the AS/400. For more information, see section 13.4, "Available On-line Help." |
| `F16` | *Major Commands* displays the first of a series of menus that can lead you to the command you need. (The menu's title is `MAJOR`, so F16 is essentially a shortcut to entering `GO MAJOR` at the command line.) |
| `F24` | *More Keys* displays more function key descriptions if there is not enough room at the bottom of the screen to describe all of the current display's available function keys. Repeatedly pressing this key cycles through descriptions of the current display's available function keys. |

When you use a terminal emulation program, certain keys may be designated as the "scroll down" and "scroll up" keys. These are analogous to the Page Up and Page Down keys, respectively. This may seem a bit backwards, and can easily lead to confusion—the key that scrolls down corresponds to the Page Up key, because it scrolls the displayed text down to show you the text above it.

The AS/400 also lets you insert and delete characters at the command line the same way you would in a text editor. Pressing the Insert key turns on insert mode; everything you type is inserted at the cursor position, moving the characters to the right of the cursor further to the right.

To return to overstrike mode while using a 3270 terminal, press the key marked "Reset." The carat symbol should disappear, and newly typed text will take the place of the characters at the cursor. (When your keyboard "locks up," or refuses to accept input, the Reset key is also useful for freeing up the keyboard.) On most PCs emulating a 3270, the Insert key does the job of the 3270 keyboard's Insert key and the Escape key serves as the Reset key. Check your emulation program's documentation to make sure.

Each time you press the Delete key, it deletes the character at the cursor.

The ability to insert and delete characters is particularly useful when used in conjunction with the F9 key. To enter a series of similar commands, start by typing out the first one and pressing Enter. For each of the remaining commands, press F9 to retrieve the last one entered, make the necessary changes to turn it into the new command, and press Enter again.

## 13.2 File Names

As you'll see in section 13.3, "How Files Are Organized," files and nearly everything else on an

AS/400 are treated as objects. You rarely have to worry about the maximum length of the name of an object or of anything else; when you enter a name on an entry screen, you're filling out a field, and underscores show the field's maximum length. You'll see that object names can be up to ten characters.

Names can use letters of the alphabet (case doesn't matter—lower case is converted to upper case) and numeric digits, and the characters $, #, @, and the period (.). Don't start a name with a numeric digit or a period.

## 13.2.1 Wildcards

There is only one wildcard in OS/400: the asterisk. It is used to designate "generic object names," which are essentially the same as the use of a wildcard to refer to multiple files at once on any other operating system. The string "abc*" refers to all applicable objects that begin with the letters "abc."

The asterisk must go at the end of a generic name. Remember, a name that starts with an asterisk is very different from a generic name—it is a predefined value, which is a special string that represents a specific value defined by the operating system.

A typical command that can make use of generic names is the Work with Members Using the Program Development Manager (`WRKMBRPDM`) command. This displays the components of a file known as "members" on a list display so that you can view and manipulate individual members. (You'll see more about this command in section 14.1.2.1, "Listing a File's Members.") If the 93 members of the `QCSRC` file are the C source code for 93 different programs, then entering the command

```
WRKMBRPDM FILE(QCSRC) MBR(*ALL)
```

lists all 93 members on the Program Development Manager display. (Since the predefined value `*ALL` is the default value for the `WRKMBRPDM` command's `MBR` parameter, it doesn't need to be included in this command.) You can enter a member name as the value for the `MBR()` parameter; entering the command

```
WRKMBRPDM FILE(QCSRC) MBR(CTEST1)
```

tells `WRKMBRPDM` to only list the member named `CTEST1`.

A generic name is a compromise between entering `*ALL` amd entering a specific member name. Entering

```
WRKMBRPDM FILE(QCSRC) MBR(CTEST*)
```

tells `WRKMBRPDM` to list the members of the `QCSRC` file that begin with the letters "CTEST." With this command, the members `CTEST1, CTEST2, CTEST3,` and `CTEST3A` will all show up in the list display.

The `GO` command is another that accepts generic names as parameters. Because there are three different menus that begin with the letters "MA," entering the command

```
GO MA*
```

displays a list with the names and descriptions of these three menus, letting you pick the one you want.

## 13.3 How Files Are Organized

OS/400 treats everything as objects. This includes files, screens, commands, terminals, databases, programs, queues, and libraries.

Each *library* holds a group of related objects. For example, there is one library for each user ID, several for the operating system, one or more for each installed application, and so forth. If the concept of a library sounds similar to the concept of a directory on other operating systems, you're right, but there is one crucial difference: the QSYS library (one of the operating system libraries) is the only one that can contain other libraries. You cannot create libraries within your libraries the way you can create subdirectories on UNIX, VMS, and DOS. As a general rule, only the system administrator can create new libraries, whether they are for new users or for new application software.

Each object has an object type. This identifies what the object is and the operations that you can perform on it. When you list the objects in a library, you will see the type of each object listed with it; the following are some common object types:

| | |
|---|---|
| `*LIB` | A library. |
| `*FILE` | A file. |
| `*CMD` | A command. |
| `*PGM` | A program. |

Certain object types have an attribute (sometimes known as the "extended attribute") that describes their role more specifically than their type name does. For example, a *PGM object has an attribute describing the language in which the program was written, such as C or RPG. For *FILE objects, the attribute plays a crucial role in identifying a particular file's structure and purpose, as you'll see in section 13.3.1, "Physical, Source Physical, and Logical Files."

When you refer to an object, you often have to identify the library where it can be found. A popular shorthand way to refer to an object's full name is `LIBNAME/OBJNAME`, where `LIBNAME` is the library name and `OBJNAME` is the object name. When the library name is included with the object name, it's known as a "qualified object name." (If an object's library is in your library list, specifying the library name will probably be unnecessary. For more on library lists, see section 13.3.2, "The Library List and Your Current Library.")

Technically, the term "file" has a more specific meaning on the AS/400 than it does on other operating systems: It is an object in a library that contains data or source code for programs. Programs and files are two different object types, so a compiled, executable program is not considered to be a file the way that it is on other operating systems.

To confuse you even further, a file can be composed of units known as *members*, which are individually comparable to a single file of source code on other operating systems. For example, a file of C source code can have multiple members, each of which might be the C source code for a different program. Members don't count as objects in the AS/400's object-oriented scheme of things, because they can't exist on their own; each member is part of a *FILE object.

Like a group of C programs on another operating system, the members of a given file generally have a similar purpose and format. In addition to storing the source code for several different programs written in the same language, a file could hold a group of data file descriptions, a group of operating system command language files, or the information necessary to display a group of menus.

If you can get used to the AS/400's fairly restricted use of the term "file," then its system of file organization—excuse me, object organization—is not really that confusing. To recap: everything is an object. Files and other objects are stored in libraries (which are also objects). A file can be subdivided into groups called members. A group of members in the same file usually have the same format and purpose, similar to a group of files with the same file extension in UNIX or DOS or files with the same file type in VM/CMS or VMS. The whole arrangement will look especially familiar to MVS users, who will recognize a strong resemblance to the concept of partitioned data sets.

### 13.3.1 Physical, Source Physical, and Logical Files

A *physical file* is a file that holds database data. This has broader applications on the AS/400 than a traditional database file does on other systems; while you might think of a database file as holding columns of data for a database (for example, an employee's last name, first name, social security number, and hire date) on the AS/400 it might hold a simple text file. Technically, such a file is still columns of data—one for line numbers, one for the date that each record (that is, each line of the text file) was last changed, and one for the line of text itself. To see an example of a physical file that holds paragraphs of text, use the `DSPPFM` command described in section 14.1.3, "Displaying a Text File's Contents," to look the `AAAMAP` member of the `QATTINFO` file in the `QUSRTOOL` library. (This IBM-supplied file describes application development and system management tools available on the system for more advanced users.)

A physical file whose members contain source code for programs, screens, or databases is known as a *source physical file*. A particular source physical file's members might be source code for several programs written in the same language. (Certain file-naming conventions make it easier to recognize the programming language by a file's name; for example, `QCSRC` is a common name for a source physical file of C code, and `QCBLSRC` is a typical name for a file of COBOL source code.) The name of the OS/400 text editor reveals its important role in creating the members of these files: the "Source Entry Utility," or SEU, is the text editor used to create and edit these files.

A *FILE object has an attribute that describes what kind of file it is. Two common values for this attribute are PF (physical file) and LF (logical file). A logical file doesn't hold database data; it holds information about an alternate format for viewing a particular physical data file or group of files. For example, if a physical file of employee data holds names, salaries, and phone numbers, a programmer might define a logical file that only shows the names and phone numbers from the physical file. A system administrator could then grant wider access to the logical file, allowing people to look up each other's phone numbers without seeing each other's salaries.

An object of `*FILE` can have several other possible attributes—for example, PF38 denotes a physical file moved from a System/38—but PF and LF are the most common.

### 13.3.2 The Library List and Your Current Library

When you want to run a program or use a data file, you don't always have to specify the library where it is located. How does OS/400 know where to find it? By searching the library list. The library list is a list of libraries that tell OS/400 where to look for objects and in what order.

Each user has their own library list. Sometimes, to use a new application program on your system, you might be told to add the library with that application's objects to your library list. (For information on doing this, see section 14.1.7, "Editing Your Library List.")

The current library is the first library where the system looks when you request the use of an object—in other words, the first library on your library list. (Actually, it's the first library on the part of the list that you can change.) Your default current library will probably be the personal library assigned for your user ID. This way, you get quick access to objects that you create.

If this book shows you an OS/400 command with `FILE(filename)` as a parameter, it usually takes it for granted that that file is in a library in your library list. If not, you must tell OS/400 where to find the file by either entering the library name as a separate parameter or by using the `FILE(libname/filename)` notation.

You can change your current library by indicating a particular library name in the "Current library" field of the signon screen or by using the `CHGCURLIB` command. For more information, see section 14.1.7, "Editing Your Library List," and section 14.1.7.1, "Changing Your Current Library."

## 13.4 Available On-line Help

OS/400 offers many kinds of help. The most important way to get help is also the most basic: press F1. (On some keyboards, there may be an actual key labelled "Help.")

This doesn't bring up some vague help menu that forces you to search through a dozen screens to find the information that you need, as F1 does with some other systems and application programs. The OS/400 on-line help is context-sensitive, so F1 often takes you right to the information you need.

All you have to do is move your cursor to the appropriate area of the screen before pressing F1. For example:

- If you move your cursor to the lines at the bottom of the screen that list the available function keys, pressing F1 displays explanations of those function keys.
- On a menu display, pressing F1 while your cursor is positioned on one of the menu choices displays information about that menu choice.
- On an entry display, pressing F1 while your cursor is in one of the fields displays "field help," or information about that field.
- In a list display, pressing F1 while you cursor is in one of the columns displays help about the information in that column.
- With your cursor on an error message, F1 displays an explanation of the cause of the error message and how to fix it.
- If you enter a command or menu choice number at the command line but press F1 instead of Enter, OS/400 displays help about that command or menu choice.

Figure 13.5 shows the screen displayed after pressing F1 with your cursor on the `MAIN` menu's function key list.

```
MAIN                              AS/400 Main Menu
..............................................................................
:                               Function Keys - Help                         :
:                                                                            :
:  F1=Help                                                                   :
:      Provides additional information about using the display or a          :
:      specific field on the display.                                        :
:                                                                            :
:  F3=Exit                                                                   :
:      Ends the current task and returns to the display from which the task  :
:      was started.                                                          :
:                                                                            :
:  F4=Prompt                                                                 :
:      Provides assistance in entering or selecting a command.               :
:                                                                            :
:  F9=Retrieve                                                               :
:                                                                  More...  :
:  F2=Extended help   F3=Exit help   F10=Move to top      F11=Search index   :
:  F12=Cancel         F13=User support   F14=Print help   F20=Enlarge        :
:                                                                           :
:...........................................................................:
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=User support
```

**Figure 13.5 Help screen for the Main Menu's function keys.**

If your cursor is on a display's command line, title, a blank line, or any part of the screen for which no specific help is available, OS/400 displays extended help. Extended help is a general description of the current display and how to use it. It's also available when you are viewing any other kind of help by pressing F2. Figure 13.6 shows the extended help for the `SNDMSG` (Send Message) command.

```
                             Send Message (SNDMSG)
 ............................................................................
 :                           Send Message - Help                            :
 :                                                                          :
 :      The Send Message (SNDMSG) command is used by a display station user  :
 :      to send an immediate message from his display station to one or more :
 :      message queues.  (An immediate message is a message that is not      :
 :      predefined and is not stored in a message file.)  The message can be :
 :      sent to the system operator, to other display station users, to a    :
 :      user's message queue, all currently active users' message queues or  :
 :      to the system history log, QHST.  The sender can require a reply     :
 :      from the message receiver.  The primary users of this command are    :
 :      display station users and the system operator.                       :
 :                                                                          :
 :          Note:  Do not precede an entry with an asterisk unless that      :
 :          entry is a "special value" that is shown (on the display itself  :
 :          or in the help information) with an asterisk.                    :
 :                                                                          :
 :  Message text (MSG)                                                       :
 :                                                                  More...  :
 :  F3=Exit help       F10=Move to top   F11=Search index   F12=Cancel       :
 :  F13=User support   F14=Print help                                        :
 :                                                                          :
 :..........................................................................:
```

**Figure 13.6 SNDMSG extended help screen.**

### 13.4.1 The Search Index

With many help systems, you can't find out about the use of a command unless you already know the command's name. The OS/400 Search Index makes it easy to find help on a topic when you only have a vague idea of the information you need and don't yet know the OS/400 name or terminology associated with that command.

To start the Search Index, either enter `STRSCHIDX` at any command line or press F11 when the bottom of a help screen tells you that `F11=Search Index`. This brings up the Search Help Index display, where you enter a word or phrase that describes the topic in question. Figure 13.7 shows the Search Help Index display.

```
                                Search Help Index

  Index Search allows you to tell the system to search for specific
  information.  To use Index Search, do the following:

    1.  Type the phrase or words to search for.

    2.  Press Enter.

  When you press Enter, the system searches for topics related to the
  words you supplied and displays a list of topics found.

  If you press Enter without typing anything, the system displays a list
  of all available topics.




  Type words to search for, press Enter.
   __________________________________________________________________________

  F3=Exit help   F5=All topics   F12=Cancel   F13=User support
```

**Figure 13.7 Search Help Index display.**

After you enter a search phrase, press Enter. OS/400 displays a list of the relevant help topics. For a vague search phrase, there may be too many topics to fit on one screen. Figure 13.8 shows the result of entering "programming" as the search phrase; note how the message `More...` in the lower right tells you that you must press Page Down (or Scroll Up) to look through the remaining topic titles. Topics are listed alphabetically, and since the list shown in Figure 13.8

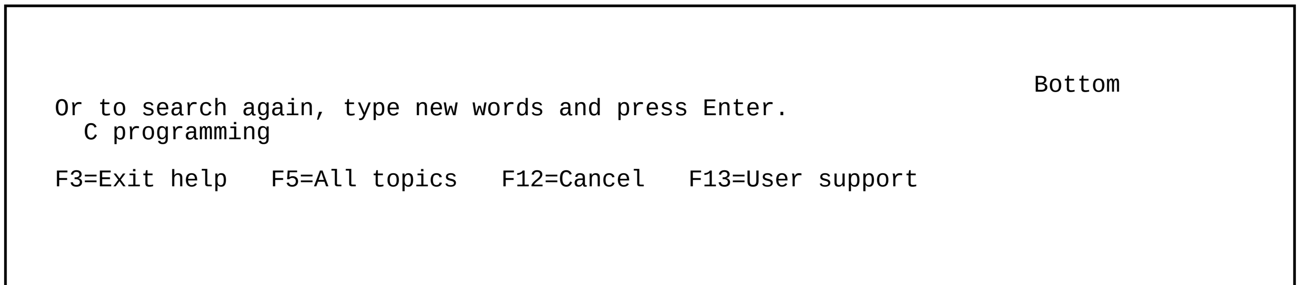
doesn't even get to the topics beginning with the letter "C," it must be quite a long list.

```
                           Main Help Index for AS/400

 Type options, press Enter.
   5=Display topic   6=Print topic

 Option  Topic
   _     Add physical file variable-length member (ADDPFVLM) command
   _     Add program (ADDPGM) command
   _     Analyze program (ANZPGM) command
   _     APPC
   _     Application program
   _     Apply program temporary fix (APYPTF) command
   _     APPN
   _     Attention-key-handling program
   _     Automatic link and external reference
   _     BASIC
   _     Branch instruction
   _     Breakpoint program

                                                                    More...
 Or to search again, type new words and press Enter.
   programming

 F3=Exit help   F5=All topics   F12=Cancel   F13=User support
```

**Figure 13.8 Search Help Index topics for "programming."**

Figure 13.9 shows the result of a search on a more specific topic, "C programming." The word "Bottom" in the lower-right shows that all the topics that fit this search phrase are displayed on this one screen.

```
                           Main Help Index for AS/400

 Type options, press Enter.
   5=Display topic   6=Print topic

 Option  Topic
   _     C language
   _     C language interface (Query Management)
   _     Create C locale description (CRTCLD) command
   _     Create C/400 program (CRTCPGM) command
   _     Create Structured Query Language C (CRTSQLC) command
   _     Delete C locale description (DLTCLD) command
   _     Including SQLCA in C
   _     Retrieve C locale description source (RTVCLDSRC) command
```



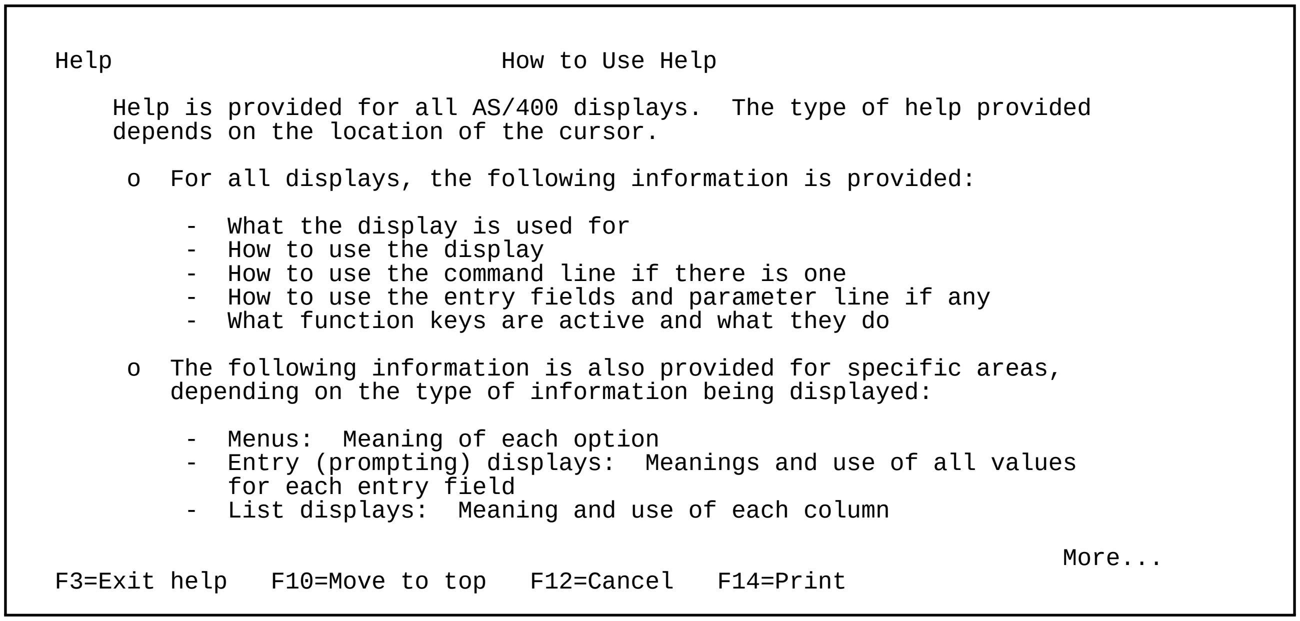
```
                                                                   Bottom
 Or to search again, type new words and press Enter.
   C programming

 F3=Exit help   F5=All topics   F12=Cancel   F13=User support
```

**Figure 13.9 Search Help Index topics for "C programming."**

Because help topics are shown on a list display, the upper part of the screen shows the numbers that you can type in the `Option` column on the left to see the help topics. A 5 displays the help information for a given topic, and a 6 sends that information to the printer.

## 13.4.2 Navigating Help Screens

The important function keys in on-line help are consistent with the rest of OS/400: Page Up (or Scroll Down) and Page Down (or Scroll Up) to page through the help text, F3 to exit help, and F12 to Cancel (that is, to back up one screen). As with other situations, F1 gives you help about what you are doing—in this case, using help. Figure 13.10 shows the first "How to Use Help" screen that OS/400 displays when you press F1 while viewing any other help screen.



```
 Help                           How to Use Help

     Help is provided for all AS/400 displays.  The type of help provided
     depends on the location of the cursor.

      o  For all displays, the following information is provided:

          -  What the display is used for
          -  How to use the display
          -  How to use the command line if there is one
          -  How to use the entry fields and parameter line if any
          -  What function keys are active and what they do

      o  The following information is also provided for specific areas,
         depending on the type of information being displayed:

          -  Menus:  Meaning of each option
          -  Entry (prompting) displays:  Meanings and use of all values
             for each entry field
          -  List displays:  Meaning and use of each column

                                                                     More...
 F3=Exit help   F10=Move to top   F12=Cancel   F14=Print
```

**Figure 13.10 The first "How to Use Help" screen.**

### 13.4.2.1 Expanding Help Windows

Sometimes, when viewing field help after pressing F1 with your cursor on a particular entry screen field, the help information appears in a window that takes up only part of your screen. Figure 13.11 shows the help displayed by pressing F1 when the cursor is in the "Send Message" entry screen's "To user profile" recipient field.

```
                             Send Message (SNDMSG)

 Type choices, press Enter.

 Message text . . . . . . . . . .






 To user profile  . . . . . . . .                 Name, *SYSOPR, *ALLACT...
               ...............................................................
               :                To user profile (TOUSR) - Help               :
               :                                                             :
               :  Specifies that the message is to be sent to the message    :
               :  queue specified in the user profile for the user named on  :
               :  this parameter.  This parameter cannot be used if a value  :
               :  is specified for the To message queue prompt (TOMSGQ       :
               :                                                    More...  :
               :  F2=Extended help   F10=Move to top   F11=Search index      :
 F3=Exit   F4= :  F12=Cancel         F20=Enlarge       F24=More keys         :
 F13=How to us :                                                             :
 Parameter MSG :.............................................................:
```

**Figure 13.11 "To user profile" field help overlaying "Send Message" screen.**

If a help window covers only part of your screen but has too much information to fit in the help window, the F20 (Enlarge) key expands it to take up the whole screen. Figure 13.12 shows the "To user profile" help expanded to fill up the whole screen. Much more information fits in this help window, so less paging will be necessary to read the whole thing.

```
                             Send Message (SNDMSG)
 ..............................................................................
 :                       To user profile (TOUSR) - Help                       :
 :                                                                            :
 :  Specifies that the message is to be sent to the message queue specified   :
 :  in the user profile for the user named on this parameter.  This           :
 :  parameter cannot be used if a value is specified for the To message       :
 :  queue prompt (TOMSGQ parameter).                                          :
 :                                                                            :
 :  Either this parameter or the To message queue prompt (TOMSGQ parameter)   :
 :  is required.                                                              :
 :                                                                            :
 :  user-profile-name                                                         :
 :      Specify the user profile name of the user to whom the message is      :
 :      sent.                                                                 :
 :                                                                            :
 :  *SYSOPR                                                                   :
 :      The message is sent to the system operator message queue,            :
 :      QSYS/QSYSOPR.                                                         :
 :                                                                   More...  :
 :  F2=Extended help   F3=Exit help   F10=Move to top   F11=Search index      :
 :  F12=Cancel         F13=User support   F14=Print help                      :
 :                                                                            :
 :............................................................................:
```

**Figure 13.12 "To user profile" field help expanded to whole screen.**

### 13.4.2.2 The User Support and Education Menu

The User Support and Education menu offers another form of access to the various kinds of on-line help. You can display it by entering `GO SUPPORT` at any command line or by pressing F13 while viewing any screen where F13 means "User support." (This applies to nearly all OS/400 menus.)

```
 SUPPORT                    User Support and Education
                                                            System:   NEPAS4
 Select one of the following:

      1. How to use help
      2. Search system help index
      3. How to use commands
      4. Question and answer
      5. AS/400 publications
      6. IBM product information
      7. How to handle system problems
      8. Problem handling
      9. Online education






 Selection or command
 ===>

 F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F16=AS/400 Main menu
 (C) COPYRIGHT IBM CORP. 1980, 1992.
```

**Figure 13.13 User Support and Education menu.**

Figure 13.13 shows an example of the User Support and Education menu. It offers the following choices:

1. *How to use help* displays the same screen that you get when you press F1 while viewing a help screen: an overview of how to use the on-line help.

2. *Search system help index* displays the Search Help Index screen, which lets you tell OS/400 to list help topics related to a word or phrase that you enter. See section 13.4.1, "The Search Index," for more information.

3. *How to use commands* displays an overview of the use of the OS/400 command line and the structure of the available commands and their parameters.

4. *Question and answer* gives you access to a database of commonly asked questions about the AS/400 and their answers. Some AS/400 systems are hooked up to mainframes at IBM sites, giving them access to a wider variety of questions and answers (an excellent example of the AS/400's potential role in a large, distributed database).

5. *AS/400 publications* displays a catalog of publications that you can order from IBM that cover OS/400, the AS/400, and available IBM development and application software.

6. *IBM product information* displays information about software available for the AS/400.

7. *How to handle system problems* displays information for system administrators about ways to approach problems with their AS/400. (Although I wonder about the paragraph on the first screen that begins "If you cannot turn the system on..." If someone can't turn the system on, they'll have a tough time taking advantage of the information provided by "How to handle system problems.")

8. *Problem handling* displays information for system administrators about the use of tools such as the job log (a history of a user's activity since signing on) and various kinds of diagnostic software used to handle problems in communications, programming development, and other aspects of the AS/400.

9. *Online education* leads you to the on-line tutorials about the AS/400 and its software. A variety of tutorials may be installed; if there is only one, it will be Tutorial System Support (TSS), which gives you background in the concepts and usage of OS/400. See section 13.4.3, "The On-line Tutorial," for more information.

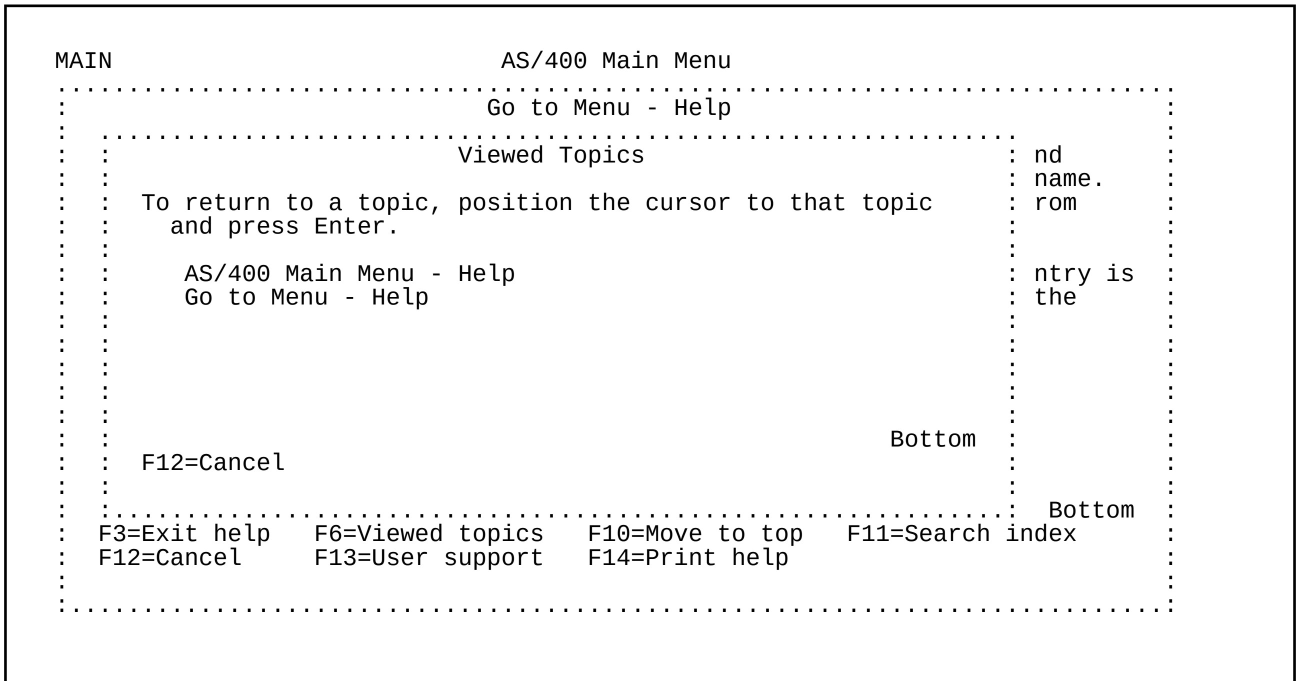
Choices four through eight on this menu are part of something called "Electronic Customer Support," or "ECS." In much of the computer industry, "customer support" means calling up some company, being put on hold, asking your question, being put on hold again, and then being told that someone will call you back with the answer. Electronic Customer Support takes advantage of the computing and communications resources at your disposal to offer you a much more efficient alternative.

#### 13.4.2.3 Hypertext Links

On many help screens, you will see certain phrases highlighted or underlined, depending on the type of terminal that you are using or emulating. This highlighting tells you that the phrase is a hypertext link to more detailed information on that phrase. These phrases may be in the middle of a sentence or in a list of topics at the end of a help topic.

For example, on the help screen that you see when you press F1 while viewing the OS/400 Main Menu, one paragraph begins with the sentence "To go to another menu, use the Go to Menu (GO) command." The phrase "Go to Menu (GO) command" is highlighted, showing that if you press your Tab key to move your cursor there and then press Enter (not F1 again—remember, F1 while viewing a help screen gives you help about the use of on-line help) you'll jump directly to a screen that tells you about the GO command.

After you use a hypertext link to another screen, a new function key becomes available: F6 (Viewed Topics). This displays a window with a list of the titles of the help screens you have viewed. You can jump directly back to any of these help screens by moving your cursor to the title of your choice and pressing Enter. Figure 13.14 shows the short list of topics displayed if you press F6 after using the hypertext link described above to go from the Main Menu's help screen to the GO command's help screen.



```
MAIN                              AS/400 Main Menu
:..............................................................................:
:                             Go to Menu - Help                                :
:  :..............................................................:            :
:  :                        Viewed Topics                         : nd         :
:  :                                                              : name.      :
:  :  To return to a topic, position the cursor to that topic     : rom        :
:  :    and press Enter.                                          :            :
:  :                                                              :            :
:  :     AS/400 Main Menu - Help                                  : ntry is    :
:  :     Go to Menu - Help                                        : the        :
:  :                                                              :            :
:  :                                                              :            :
:  :                                                              :            :
:  :                                                              :            :
:  :                                                     Bottom   :            :
:  :  F12=Cancel                                                  :            :
:  :                                                              :            :
:  :..............................................................:  Bottom    :
:  F3=Exit help   F6=Viewed topics   F10=Move to top   F11=Search index        :
:  F12=Cancel     F13=User support   F14=Print help                            :
:                                                                              :
:..............................................................................:
```

**Figure 13.14 Viewed Topics window listing help screens displayed with hypertext links.**

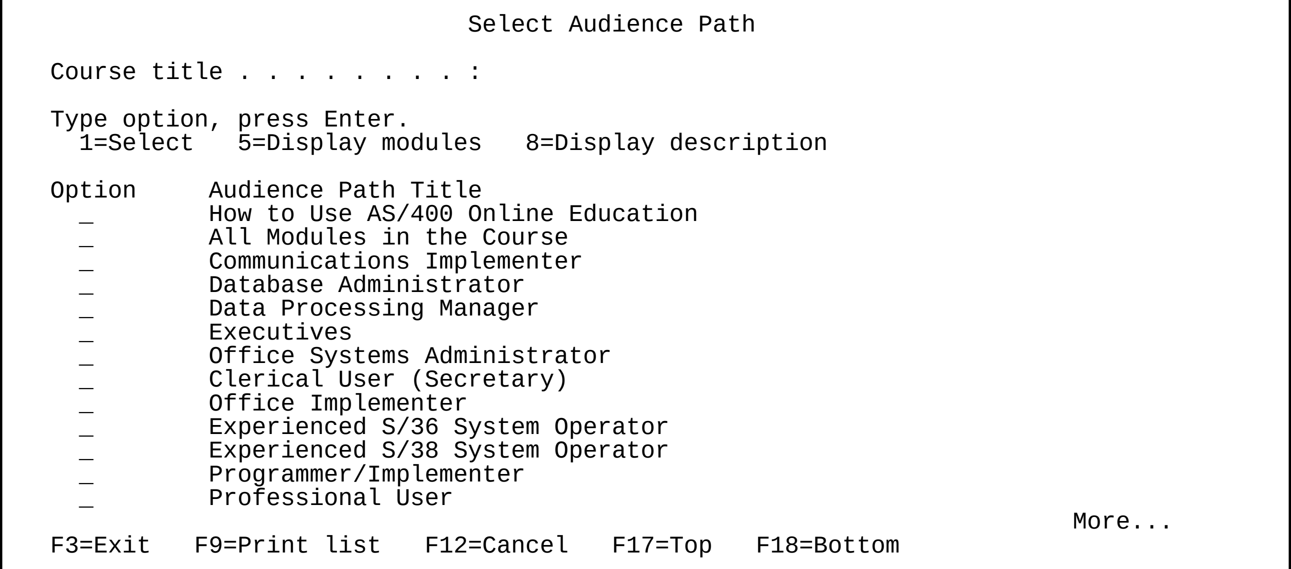
## 13.4.3 The On-line Tutorial

OS/400 offers various on-line tutorials to help you learn about the system. They let you work at your own pace, with exercises, reviews, and quizzes. Each course is broken down into lessons known as "modules" that can be completed in 15 to 40 minutes. Each module warns you at the beginning how long it will take. The system keeps track of which modules you have taken of which courses, which makes it easier to pick up where you left off if you haven't worked on a particular tutorial in a while. You can even leave "bookmarks" of where you were if you have to leave a module before finishing it.

The most important course is Tutorial System Support (TSS), which gives you background in the concepts and usage of OS/400. Even if many tutorials are available on your system, this is the best one to start with.

To start the on-line education, you can select Online Education from the "User Support and Education" menu described in section 13.4.2.2 or you can enter `STREDU` (Start Education) from the command line.

The first time you do this, an entry display asks you for your first and last name. The system uses this information to remember which courses you have taken. Next, a list display asks you to pick from a list of available courses. After you pick one, another list asks you to pick an audience path, as shown in Figure 13.15.



```
                             Select Audience Path

 Course title . . . . . . . . :

 Type option, press Enter.
   1=Select   5=Display modules   8=Display description

 Option     Audience Path Title
   _        How to Use AS/400 Online Education
   _        All Modules in the Course
   _        Communications Implementer
   _        Database Administrator
   _        Data Processing Manager
   _        Executives
   _        Office Systems Administrator
   _        Clerical User (Secretary)
   _        Office Implementer
   _        Experienced S/36 System Operator
   _        Experienced S/38 System Operator
   _        Programmer/Implementer
   _        Professional User
                                                                       More...
 F3=Exit   F9=Print list   F12=Cancel   F17=Top   F18=Bottom
```

**Figure 13.15 List display asking you to pick an audience path for an on-line course.**

The "audience path" concept is a nice design touch. Of the many modules available for each course, certain combinations have been grouped into "paths" for different audiences. For example, Figure 13.15 shows different paths for database administrators, secretaries, executives, and programmers. While all four of these paths have the modules "Getting Started with Online Education" and "Working with System Displays," only the database administrator and programmer paths have the modules "Object Management Concepts." Note also the path "All Modules in the Course"—while time-consuming, this would obviously give you the most detailed background on the use of the AS/400.

Once you've selected a path, the "Select Course Option" menu offers you the following choices:

- Start the next module in your path.

- Select a different module.
- Return to the displays where you picked your course or your path.

Once you start a module, the on-line education leads you through the step-by-step tutorial.

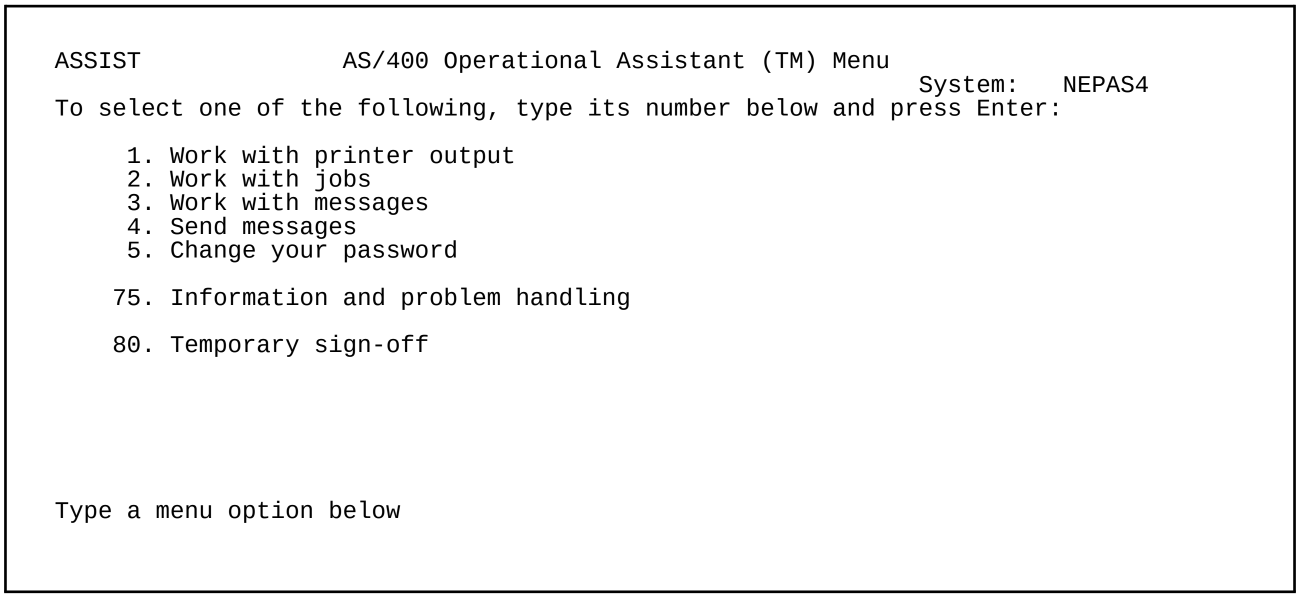
## 13.4.4 Other Helpful Features

OS/400 includes several other features that, while not actually part of on-line help, make it much easier to navigate and to use the system.

The Display Keyboard Map command (`DSPKBDMAP`) displays a series of help screens that describe the use of the keys on the terminal that you are using or emulating. If an OS/400 screen refers to a key that isn't on your keyboard, use `DSPKBDMAP` to find out which of your keyboards's keys are doing the unfamiliar key's job.

OS/400 enables you to do so much by using menus that some menus can overwhelm many users. A useful feature called the Operational Assistant can ease this problem. The Operational Assistant is a set of simple menus with a minimum of fancy terminology that let you find the most important tasks quickly and easily. In addition to allowing the basic tasks required by all users, it enables a system administrator to add new users, to backup the hard disk to tape, and to do other tasks that are crucial to running the system. This kind of feature makes the AS/400 one of the most "plug and play" large computers ever available, because it allows novice AS/400 users to set up the computer and become productive very quickly with a minimum of technical assistance.

To see the first Operational Assistant menu, enter `GO ASSIST`. Figure 13.16 shows this menu.


```
ASSIST              AS/400 Operational Assistant (TM) Menu
                                                            System:   NEPAS4
To select one of the following, type its number below and press Enter:

     1. Work with printer output
     2. Work with jobs
     3. Work with messages
     4. Send messages
     5. Change your password

    75. Information and problem handling

    80. Temporary sign-off




Type a menu option below
```

```
 F1=Help   F3=Exit   F9=Command line   F12=Cancel
```

**Figure 13.16 Operational Assistant main menu.**

# Chapter 14 Using Files in OS/400

## 14.1 The 12 Most Important Commands

OS/400 seems to have a larger number of crucial commands than other operating systems because of the file/member system. You want the ability to perform the basic operations on files and on members within files. This doesn't really require memorizing more commands, because the OS/400 menus and on-line help make it easy to find the commands whose names you can't remember.

The twelve most important commands in OS/400 are:

| | |
|---|---|
| `DSPLIB` | lists the files (and other objects) in a library. |
| `DSPFD` | lists the members in a file. |
| `DSPPFM` | displays a file or file member's contents. |
| `CRTDUPOBJ` | copies files and other objects. |
| `CPYSRCF` | copies file members. |
| `RNMOBJ` | renames files and other objects. |
| `RNMM` | renames file members. |
| `DLTF` | deletes files. |
| `RMVM` | deletes file members. |
| `EDTLIBL` | edits your library list. |
| `CRTLIB` | creates libraries. |
| `DLTLIB` | deletes libraries. |

### 14.1.1 Common Error Messages

Remember, the greatest thing about OS/400 error messages is that you can always move your cursor to them and press F1 to display an explanation of what they mean if a message is too cryptic. For example, let's look at the error message that you get when you pick a non-existent menu choice. After you enter `75` at the command line while viewing the main menu, OS/400 responds with the error message shown at the bottom of Figure 14.1.

```
MAIN                          AS/400 Main Menu
                                                            System:   NEPAS4
Select one of the following:

     1. User tasks
     2. Office tasks
     3. General System Tasks
     4. Files, libraries, and folders
     5. Programming
     6. Communications
     7. Define or change the system
     8. Problem handling
     9. Display a menu
    10. Information Assistant options
    11. PC Support tasks

    90. Sign off

Selection or command
===> 75

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=User support
F23=Set initial menu
Specified menu selection is not correct.
```

**Figure 14.1 Main menu with error message for incorrect menu choice.**

If you move your cursor to the line "Specified menu selection is not correct" and press F1, OS/400 displays the more detailed "Additional Message Information" about the error message, as shown in Figure 14.2.

```
                        Additional Message Information

Message ID . . . . . . :   CPD6A64
Date sent  . . . . . . :   07/10/93      Time sent  . . . . . . :   19:33:53

Message . . . . :   Specified menu selection is not correct.

Cause . . . . . :   The selection that you have specified is not correct for
  one of the following reasons:
    -- The number selected was not valid.
    -- Something other than a menu option was entered on the option line.
Recovery  . . . :   Select a valid option and press the Enter or Help key
  again.




                                                                      Bottom
  Press Enter to continue.
```

**Figure 14.2 Additional information about the "Specified menu selection is not correct" error message.**

Another common error message tells you that OS/400 can't find an object that you asked to use. If you enter a command that uses a non-existent object named "POTRZEBIE" (for example, try entering `GO POTRZEBIE`) you get an error message similar to this:

```
Object POTRZEBIE in library *LIBL not found
```

`*LIBL` is the OS/400 predefined value that refers to the library list. The error message means that the object POTRZEBIE cannot be found in any of the libraries in the library list.

`*LIBL`, as a representative of all the libraries in the library list, sometimes takes the place in error messages of a specific library name. If you had indicated a specific non-existent library when you identified the object to use (for example, `LOOBRARY/POTRZEBIE`), you might have seen that library's name instead of `*LIBL` in the error message.

The "not found" error message can be more specific about what it's looking for than just calling it an "object." For example, if you were copying a file and specified a non-existent "from file" (that is, the file that you are making a copy of, or copying "from") called `HELLFILE`, the error message would tell you the following:

```
From-file HELLFILE in *LIBL not found.
```

Entering the non-existent command `CLEANRUG` at the command prompt gives you this error message:

```
Command CLEANRUG in library *LIBL not found.
```

Let's say that the `POTRZEBIE` program does exist in the library `SUPERLIB`, but you're not authorized to use it. You'll get a fairly straightforward error message:

```
Not authorized to object POTRZEBIE in SUPERLIB type *PGM.
```

One more hint about dealing with errors: if you make a mistake when filling out a command prompt display (the screen you get when you press F4 after entering a command at the command line) OS/400 returns you to the previous screen and displays the command line version of the command on the command line with the appropriate error message underneath it. Of course you can move your cursor to the error message and press F1, but if you want some background on using that command, you can also move your cursor to the command line and press F1.

### 14.1.2 Listing File Names

On the AS/400, you'll want to list more than just file names. In addition to listing the names of a given library's files and other objects, it's also handy to be able to see the members of a given file. This is covered in section 14.1.2.1, "Listing a File's Members."

Use the Display Library command (`DSPLIB`) to list the objects in a library. (Don't confuse it with the similarly spelled Display Library List command, `DSPLIBL`—see section 14.1.7, "Editing Your Library List," for more information on this.)

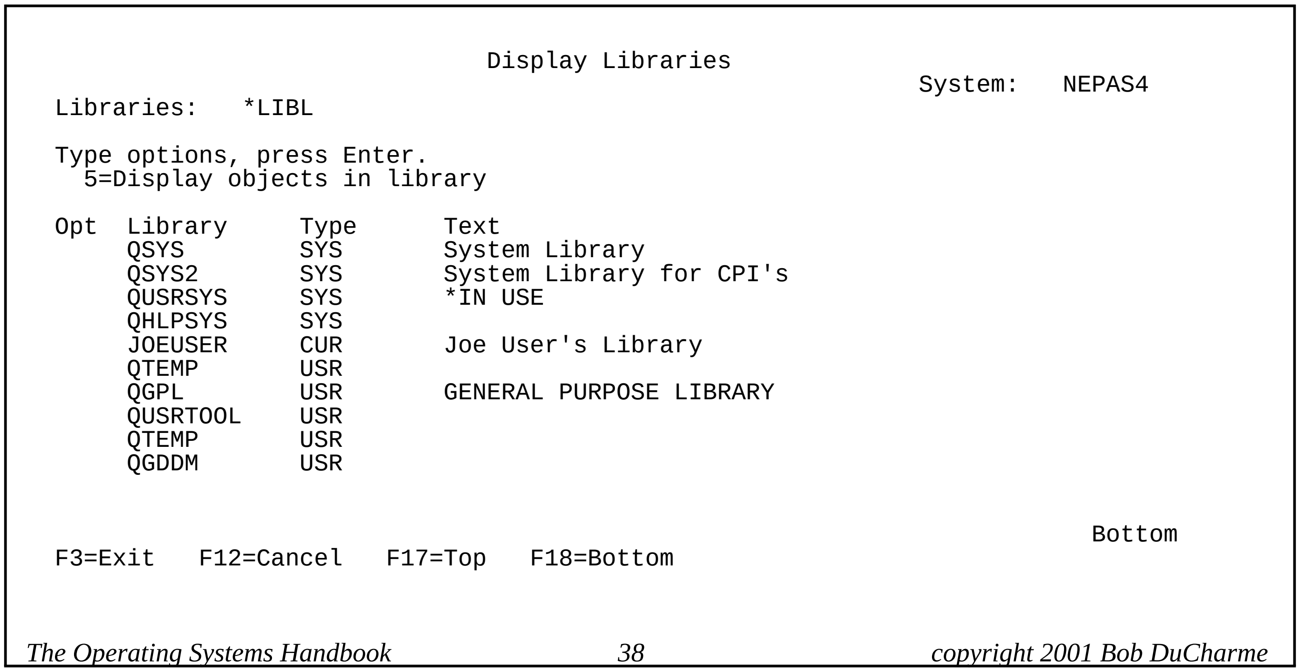
From the command line, you tell OS/400 to list the contents of a particular library by entering `DSPLIB` with the library name as the `LIB` parameter. For example, to look at the `JOEUSER` library, Joe would enter the following:

```
DSPLIB LIB(JOEUSER)
```

If you leave out the `LIB()` part, the system assumes that the `DSPLIB` command's single parameter is a library name, so the following works just as well:

```
DSPLIB JOEUSER
```

If you type `DSPLIB` with no parameters and press Enter, the system assumes a default parameter of `LIB(*LIBL)`, which tells it "I want to look at a list of objects from one of the libraries in my library list. List them out and then I'll pick one." This brings up a screen like the one shown in Figure 14.3.



```
                              Display Libraries
                                                              System:   NEPAS4
Libraries:   *LIBL

Type options, press Enter.
  5=Display objects in library

Opt  Library     Type      Text
     QSYS        SYS       System Library
     QSYS2       SYS       System Library for CPI's
     QUSRSYS     SYS       *IN USE
     QHLPSYS     SYS
     JOEUSER     CUR       Joe User's Library
     QTEMP       USR
     QGPL        USR       GENERAL PURPOSE LIBRARY
     QUSRTOOL    USR
     QTEMP       USR
     QGDDM       USR


                                                                        Bottom
F3=Exit   F12=Cancel   F17=Top   F18=Bottom
```

**Figure 14.3 Sample list of libraries displayed when you press Enter after entering DSPLIB with no parameters.**

As the display tells you, all you need to do to list the objects in one of these libraries is to enter a 5 in the `Opt` column next to a library name and press Enter. For example, entering a 5 next to `QGPL` and pressing Enter displays a screen similar to the one shown in Figure 14.4. Note the "More..." message at the bottom; `QGPL` has more objects than will fit on one Display Library screen, and you will need your Page Down and Page Up (or Scroll Up and Scroll Down, respectively) keys to see the other objects in the list.

```
                               Display Library

Library  . . . . . . :   QGPL            Number of objects  . :   208
Type . . . . . . . . :   PROD            ASP of library . . . :   1
Create authority . . :   *SYSVAL

Type options, press Enter.
  5=Display full attributes   8=Display service attributes

Opt  Object      Type      Attribute   Freed         Size  Text
     ALRC        *PGM      CLP          NO           7680  Does ADDLIBLE NEW (
     CC          *PGM      C            NO          54784  compile with as400
     CC_CMD      *PGM      C            NO          55808  compile with as400
     CCSYSTEM_   *PGM      C            NO          12288
     DKJFOCCLP   *PGM      CLP          NO          22528
     DKJSINKC    *PGM      CLP          NO          19968
     FUNCLP      *PGM      CLP          NO          22016
     FUNCLP2     *PGM      CLP          NO          20992  FUN System Developm
     INZQBATCH   *PGM      CLP          NO          11776  Pgm called when QBA
     QDCUPF      *PGM      PAS          NO          44544
     QRZHWUG1    *PGM                                   0  *NOT AUTHORIZED
                                                                      More...
F3=Exit   F12=Cancel   F17=Top   F18=Bottom
(C) COPYRIGHT IBM CORP. 1980, 1992.
```

**Figure 14.4 List of objects in the QGPL library.**

As described earlier, the commands

```
DSPLIB LIB(QGPL)
```

or

```
DSPLIB QGPL
```

would have taken you right from the command line to the list of objects in the `QGPL` library.

To the right of the `Opt` column, the other columns in the library object list tell you the following information:

| | |
|---|---|
| `Object` | The object's name. |
| `Type` | The object's type, as described in section 13.3, "How Files Are Organized." |
| `Attribute` | If applicable, the object's attribute, as described in section 13.3.1, "Physical, Source Physical, and Logical Files." |
| `Freed` | Whether the object's storage space has been freed up. This only concerns advanced users. |
| `Size` | The object's size in bytes. If you are not authorized to use the object, a zero will appear here. |
| `Text` | Descriptive text that can be added when the object is created. If the whole description does not appear on the library object list screen, entering either 5 or 8 in the `Opt` column displays the complete description along with other technical information about the object. |

According to the `Type` column in Figure 14.4, most of the objects in the `QGPL` library look like programs. A personal library would have more objects of type *FILE. For example, entering the command

```
DSPLIB LIB(JOEUSER)
```

shows the object's in Joe User's library, as shown in Figure 14.5.

```
                               Display Library

Library  . . . . . . :   JOEUSER         Number of objects  . :   31
Type . . . . . . . . :   PROD            ASP of library . . . :   1
Create authority . . :   *SYSVAL

Type options, press Enter.
  5=Display full attributes   8=Display service attributes

Opt  Object      Type      Attribute   Freed         Size  Text
     SETUP       *PGM      CLP          NO           1022
     STRIPPRN    *PGM      C            NO           3234  Strip codes from pr
     RPGTEST     *PGM      RPG          NO            894  My first RPG progra
     JOEUSER     *OUTQ                  NO           4608
     OUTPUT      *OUTQ                  NO           8704
```

```
      BUDGET93    *FILE     PF           NO          11822
      BUDGET94    *FILE     PF           NO          10432
      D2235324    *FILE     PF           NO          18944
      DEF         *FILE     PF           NO          15360
      JOETEST     *FILE     PF           NO           9216  Sample data to play
      SALES93     *FILE     PF           NO           7320
                                                                        More...
 F3=Exit   F12=Cancel   F17=Top   F18=Bottom
 (C) COPYRIGHT IBM CORP. 1980, 1992.
```

**Figure 14.5 Objects in Joe User's personal library.**

### 14.1.2.1 Listing a File's Members

The Display File Description command (`DSPFD`) will tell you more than you ever want to know about a given file. Fortunately, there is a way to tell this command "don't tell me every little technical detail about this file; just tell me the file's members." As its command prompt display tells you, it has an optional field called "Type of information." The default value is `*ALL`, but if you don't want to scroll through all the technical details to find the member list, you can enter a `TYPE` of `*MBRLIST`. If Joe User wants to enter a command at the command line that displays a list of his CL program source code members, which are in the file `QCLSRC`, he enters

```
DSPFD FILE(JOEUSER/QCSRC) TYPE(*MBRLIST)
```

or even just this:

```
DSPFD QCSRC TYPE(*MBRLIST)
```

The first screen displayed by this command shows some summary information about the file. To see the beginning of the list of member names, press Page Down (or Scroll Up) once.

Figure 14.6 shows the results of paging down once after executing this command with the `JOEUSER/QCSRC` file. As you can see, `DSPFD` shows more than just the members' names—it also shows the size, creation date, and the date and time that each member was last modified.

```
                             Display Spooled File
 File  . . . . . :   QPDSPFD                          Page/Line   1/36
 Control . . . . .                                    Columns     1 - 78
 Find  . . . . . .
 *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
   CONVERT           25322 C      07/18/93 07/18/93 16:07:37          4
     Text:  Conversion routines
   CTEST1             4096 C      07/09/93 05/16/93 09:26:36          4
     Text:  my first C/400 program
   CTEST2             6144 C      07/16/93 05/16/93 09:35:45         30
     Text:  Another C programming test
   GETDATA            4096 C      07/18/93 07/18/93 16:02:48         20
     Text:  Get data from main file
   LKNGLASS           8198 C      07/16/93 07/16/93 09:44:17          4
     Text:  Alice's Adventures Through the Looking Glass
   MAIN344            4096 C      07/18/93 07/18/93 16:04:11         20
     Text:  Module 344 of main procedure
   MAIN346            9416 C      07/18/93 07/30/93 09:23:45         20
     Text:  Module 346 of main procedure
   MAIN347           12844 C      07/18/93 07/23/93 13:36:11         20
     Text:  Module 347 of main procedure
                                                                       More...
 F3=Exit   F12=Cancel   F19=Left   F20=Right   F24=More keys
```

**Figure 14.6 The members of Joe User's QCSRC file as listed by the DSPFD command.**

### 14.1.2.2 Listing a File's Members with the Program Development Manager

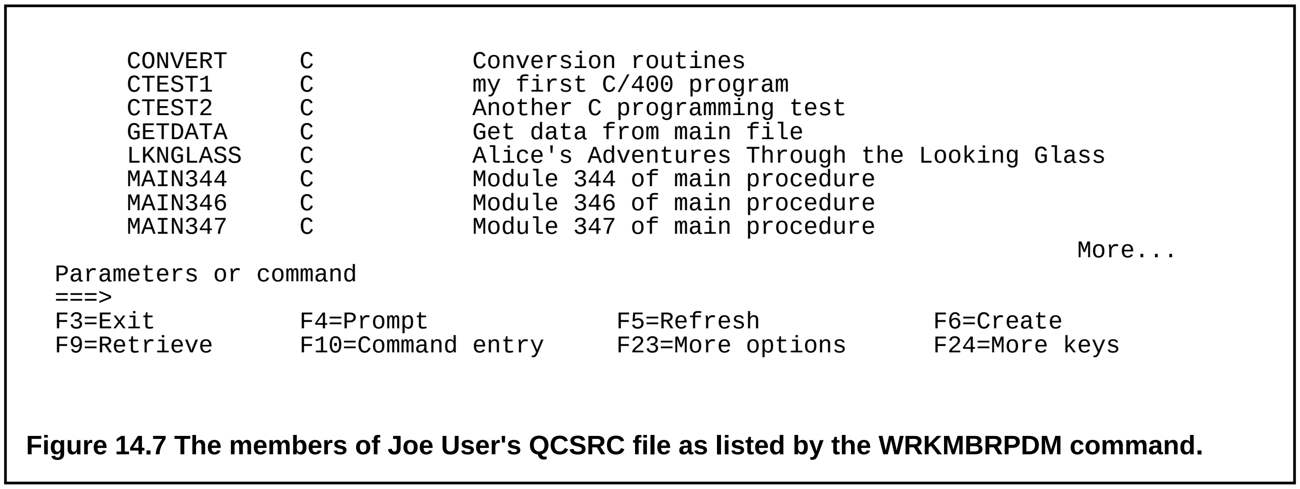
Another quick way to list a file's members is to use the Work with Members using the Program Development Manager `WRKMBRPDM` command. (Like OfficeVision, the Program Development Manager is an IBM "Licensed Program Product" or "LPP" and therefore not included as part of OS/400, so you may not have it installed on your system.)

Entering the command

```
WRKMBRPDM FILE(libname/filename)
```

displays a file's members on a list display. Because the program lets you "work with" the members, you can use this list to edit, copy, delete, display, print and rename a file's members, as we'll see later in this chapter. Figure 14.7 shows how `WRKMBRPDM` lists the members of Joe User's `QCSRC` file.

```
                            Work with Members Using PDM

 File  . . . . . .   QCSRC
   Library . . . .     JOEUSER             Position to  . . . . .

 Type options, press Enter.
   2=Edit        3=Copy       4=Delete       5=Display      6=Print
   7=Rename      8=Display description       9=Save         13=Change text ...

 Opt  Member      Type        Text
```



```
      CONVERT     C           Conversion routines
      CTEST1      C           my first C/400 program
      CTEST2      C           Another C programming test
      GETDATA     C           Get data from main file
      LKNGLASS    C           Alice's Adventures Through the Looking Glass
      MAIN344     C           Module 344 of main procedure
      MAIN346     C           Module 346 of main procedure
      MAIN347     C           Module 347 of main procedure
                                                                          More...
 Parameters or command
 ===>
 F3=Exit          F4=Prompt              F5=Refresh             F6=Create
 F9=Retrieve      F10=Command entry      F23=More options       F24=More keys
```

**Figure 14.7 The members of Joe User's QCSRC file as listed by the WRKMBRPDM command.**

Many people find `WRKMBRPDM` easier to use than `DSPFD`, but keep in mind that it is not available on all AS/400 installations.

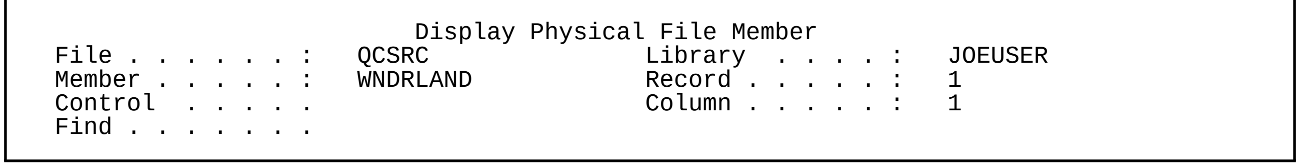
## 14.1.3 Displaying a Text File's Contents

When you want to see text stored in a file on an AS/400, you want to see a particular member of a file. Do this with the Display Physical File Member (`DSPPFM`) command. This command needs to know the name of the file and the name of the member that you want to see within that file. (It doesn't really need to know a member name; if you fail to supply one, it displays the first member of that file.)

In Joe User's `QCSRC` file, he has a C source code member called `WNDRLAND` that displays a passage from "Alice in Wonderland." To view it, he enters the following:

```
DSPPFM FILE(JOEUSER/QCSRC) MBR(WNDRLAND)
```

(If the `JOEUSER` library is in Joe's library list he doesn't really need to include the qualified object name for `QCSRC`. He could have just written `FILE(QCSRC)` as the `DSPPFM` command's first parameter.) Figure 14.8 shows the result of this command: the Display Physical File Member screen with the first 15 lines of the `WNDRLAND` program.



```
                         Display Physical File Member
 File . . . . . . :   QCSRC               Library  . . . . :   JOEUSER
 Member . . . . . :   WNDRLAND            Record . . . . . :   1
 Control  . . . . .                       Column . . . . . :   1
 Find . . . . . . .
```

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
000100930709#include <stdio.h>
000200930709main() {
000300930716   printf("Soon her eye fell on a little glass box that was lying\
000400930716   printf("under the table: she opened it, and found in it a very\
000500930716   printf("small cake, on which the words \"EAT ME\" were \n");
000600930716   printf("beautifully marked in currants.  \"Well, I'll eat it,\"
000700930716   printf("said Alice, \"and if it makes me grow larger, I can\n")
000800930716   printf("reach the key; and if it makes me grow smaller, I can\n
000900930716   printf("creep under the door: so either way I'll get into the \
001000930716   printf("garden, and I don't care which happens!\"\n\n");
001100930716   printf("She ate a little bit, and said anxiously to herself\n")
001200930716   printf("\"Which way?  Which way?\", holding her hand on the top
001300930716   printf("of her head to feel which way it was growing; and she\n
001400930716   printf("was quite surprised to find that she remained the same
001500930716   printf("size.  To be sure, this is what generally happens when\
                                                                        More...
F3=Exit   F12=Cancel   F19=Left   F20=Right   F24=More keys
```

**Figure 14.8 The first fifteen lines of the WNDRLAND program as displayed by DSPPFM.**

This file has three columns, although the first two appear as one big column. These two show the number and the date of the last change for each line. In Figure 14.8, the first two lines were created on July 9, 1993, and the rest were added a week later on the 16th.

The third column, which takes up most of the screen, is the text file itself. Lines that are too long to fit on the screen end with the slash (\) character; we'll see a way to scroll to the right to see what we're missing. (The other slashes are part of the C program. A slash before a quotation mark tells the computer that that quotation mark is part of the text that the program should output onto the screen, and a slash before the letter "n" causes a carriage return in the program's output.)

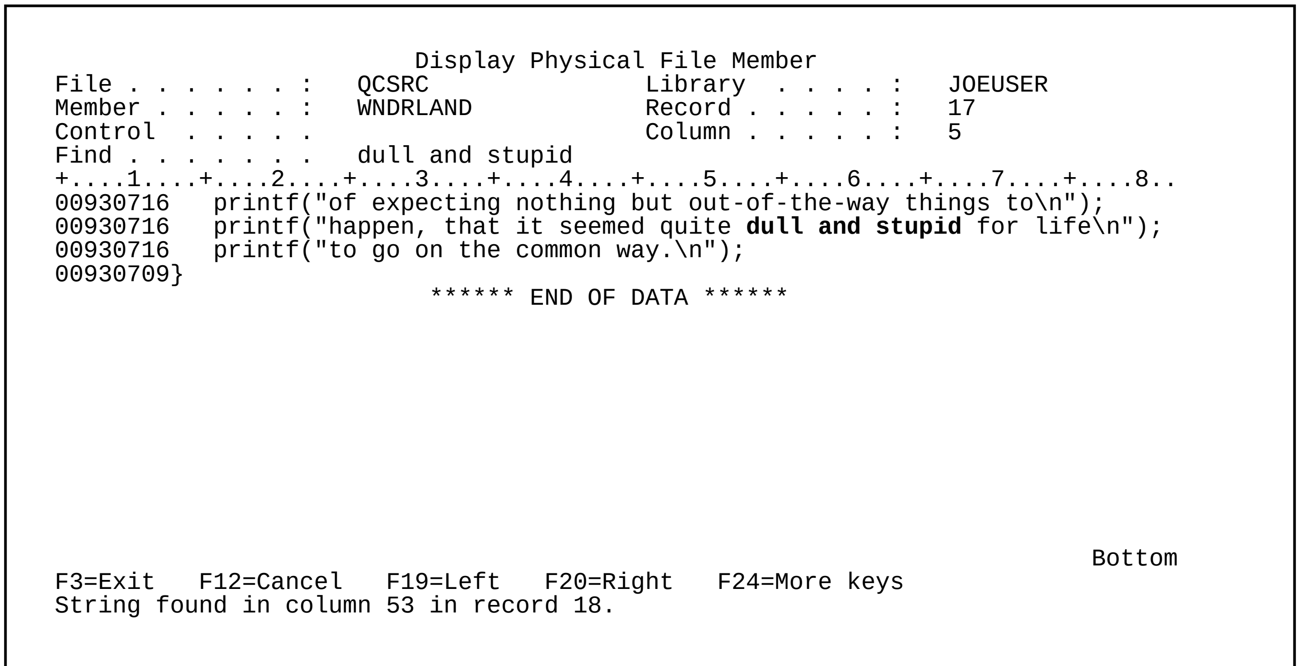
To move around in the file, the description of the function keys at the bottom of the screen shows that F19 and F20 will scroll a screenful to the left and right. Your Page Down (or Scroll Up) and Page Up (or Scroll Down) keys will have the same effect here that they have when you view anything else too long to fit on the screen.

The following commands, which you enter in the `Control` field at the top of the screen, give you additional options for scrolling text that doesn't fit on the screen:

| | |
|---|---|
| `n` | Make line `n` the top line on the screen. |
| `+n` | Move forward `n` lines. |

| | |
|---|---|
| `-n` | Move back `n` lines. |
| `W+n` | Shift the text `n` characters to the left. |
| `W-n` | Shift the text `n` characters to the right. |

The `Find` field at the top of the screen makes it easy to search for a string of text. Enter a string there, press F16, and the system puts the first line with that string near the top of the screen with the target string highlighted. (Although there was nothing at the bottom of Figure 14.8 about F16, pressing F24 for "More keys" once or twice would have shown you that F16 meant "Find.") Figure 14.9 shows the result of a search for the string "dull and stupid." Note the message at the bottom of the screen that tells the column and record (that is, line of the file) where the string was found.



```
                         Display Physical File Member
File . . . . . . :   QCSRC               Library  . . . . :   JOEUSER
Member . . . . . :   WNDRLAND            Record . . . . . :   17
Control  . . . . .                       Column . . . . . :   5
Find . . . . . . .   dull and stupid
+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8..
00930716   printf("of expecting nothing but out-of-the-way things to\n");
00930716   printf("happen, that it seemed quite dull and stupid for life\n");
00930716   printf("to go on the common way.\n");
00930709}
                          ****** END OF DATA ******










                                                                        Bottom
F3=Exit   F12=Cancel   F19=Left   F20=Right   F24=More keys
String found in column 53 in record 18.
```

**Figure 14.9 Result of searching the WNDRLAND file for the string "dull and stupid."**

When you are finished looking at the file member, press F3 to exit `DSPPFM.`

### 14.1.3.1 Displaying a File's Members with the Program Development Manager

If you use the Program Development Manager to list the names of a file's members, as shown in section 14.1.2.2, "Listing a File's Members with the Program Development Manager," you can display one of the listed members by simply entering the number 5 next to that member's name in the `Opt` column.

## 14.1.4 Copying Files

OS/400 does have a command called `CPYF` to copy files, but its most popular uses can be a bit specialized. An easier, more versatile command is "Create Duplicate Object," or `CRTDUPOBJ`.

The `CRTDUPOBJ` needs to know four things:

| | |
|---|---|
| `OBJ()` | The name of the object being duplicated. |
| `FROMLIB()` | The library where this object is stored. |
| `OBJTYPE()` | The type of the object being duplicated. Because you can have two objects with the same name in the same library (as long as they have different types) specifying the object type tells OS/400 more specifically which one you want to duplicate. If you want to copy everything in the `FROMLIB()` library with the name given by `OBJ()`, enter the predefined value *ALL as the `OBJTYPE`. To copy a file object, enter an `OBJTYPE()` of *FILE. |
| `NEWOBJ()` | The new object's name. If an object with this name already exists, OS/400 displays an error message and does not make the copy. |

When used to copy a file, `CRTDUPOBJ` copies the file and all of its members. But it doesn't necessarily copy the contents of these members—you have to tell it to do that with an additional parameter: `DATA()`. A value of *NO is the default; this tells `CRTDUPOBJ` not to copy the records in the file's members. A parameter of `DATA(*YES)` tells `CRTDUPOBJ` to copy the members' text when the file is copied.

To copy his `QCSRC` file and all the C source code in it to a file called `CCODE`, Joe User could enter the following:

```
CRTDUPOBJ OBJ(QCSRC) FROMLIB(JOEUSER) OBJTYPE(*ALL) NEWOBJ(CCODE) DATA(*YES)
```

What if Joe tries this command without the parameter names? He enters this:

```
CRTDUPOBJ QCSRC JOEUSER *ALL CCODE
```

and sees this:

```
Library CCODE not found.
```

If the fourth parameter doesn't have a parameter name, `CRTDUPOBJ` assumes that it's the name of the library to which you are copying the object—the `TOLIB()`. If you leave out the `TOLIB()`, `CRTDUPOBJ` assumes that you want to put the copy in the same library as the original. If you leave out the `TOLIB()` but want to include more than three parameters, everything after the third must therefore have a parameter name. Otherwise, the program thinks that your fourth parameter is the `TOLIB()`. In Joe's last attempt at copying `QCSRC`, `CRTDUPOBJ` saw `CCODE` in the fourth position, looked for a library by that name, and didn't find it.

Positional parameters are not an all-or-nothing proposition. You can include positional parameters without the parameter names and parameters that include the parameter names in the same command, as long as all positional parameters come before all named parameters. Joe will have no trouble if he enters the command like this:

```
CRTDUPOBJ QCSRC JOEUSER *ALL NEWOBJ(QCSRC1) DATA(*YES)
```

### 14.1.4.1 Copying Members

The Copy Source File (`CPYSRCF`) command lets you copy individual source file members. The copy can go in the same source file or to a different one.

`CPYSRCF` needs to know at least three things:

| | |
|---|---|
| `FROMFILE()` | The file with the member you are copying. |
| `TOFILE()` | The file where the copy should go. Even if the copy will go in the same file as the original, this must be included. |
| `FROMMBR()` | The name of the member being copied. |

When you copy a member from one file to another, the new member can have the same name as the old one. If your copy is going to be in the same file as the original (that is, the `TOFILE()` is the same as the `FROMFILE()`), the command will need to know something else: the name that you want to give the copy. Use the `TOMBR()` parameter for this.

If you specify an existing file member as the `TOMBR()`, `CPYSRCF` copies the `FROMMBR()` right over it with no warning.

Joe User's first C source code program on the AS/400 was the member `CTEST1` in his `QCSRC` file. He wants to try a variation on this program, but he wants to keep the original intact, so he makes a copy in the same file called `CTEST1A` with the following command:

```
CPYSRCF FROMFILE(JOEUSER/QCSRC) TOFILE(JOEUSER/QCSRC) FROMMBR(CTEST1) TOMBR(CTEST1A)
```

He can then edit `CTEST1A` to try out his new ideas.

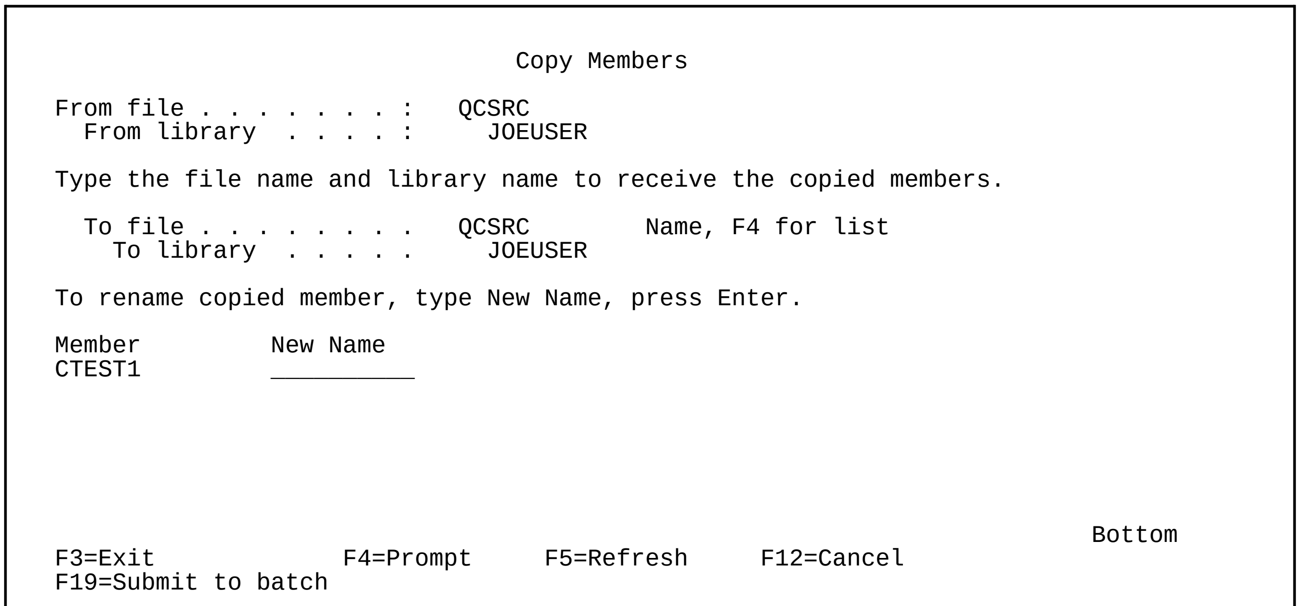
Next, Joe's `QCSRC` file has a member called `CONVERT` which has the code for several data conversion routines. He needs this in another file called `BIOINPUT` and copies it with the following command:

```
CPYSRCF QCSRC BIOINPUT CONVERT
```

Because all the parameters that he needed to enter were required, he entered them as positional parameters without their parameter names and saved himself some typing.

#### 14.1.4.2 Copying Members with the Program Development Manager

Once you've listed a file's members with the Program Development Manager, you can copy one of these members by entering the number 3 next to that member's name in the `Opt` column. The Program Development Manager then displays the screen shown in Figure 14.10 to find out the name that you want to give the new member. If you want to put the copy in a different file or library, enter their names here as well. (If you don't want to put it in a new file or library, note that the FROMFILE's file and library names are already entered as the default values.)



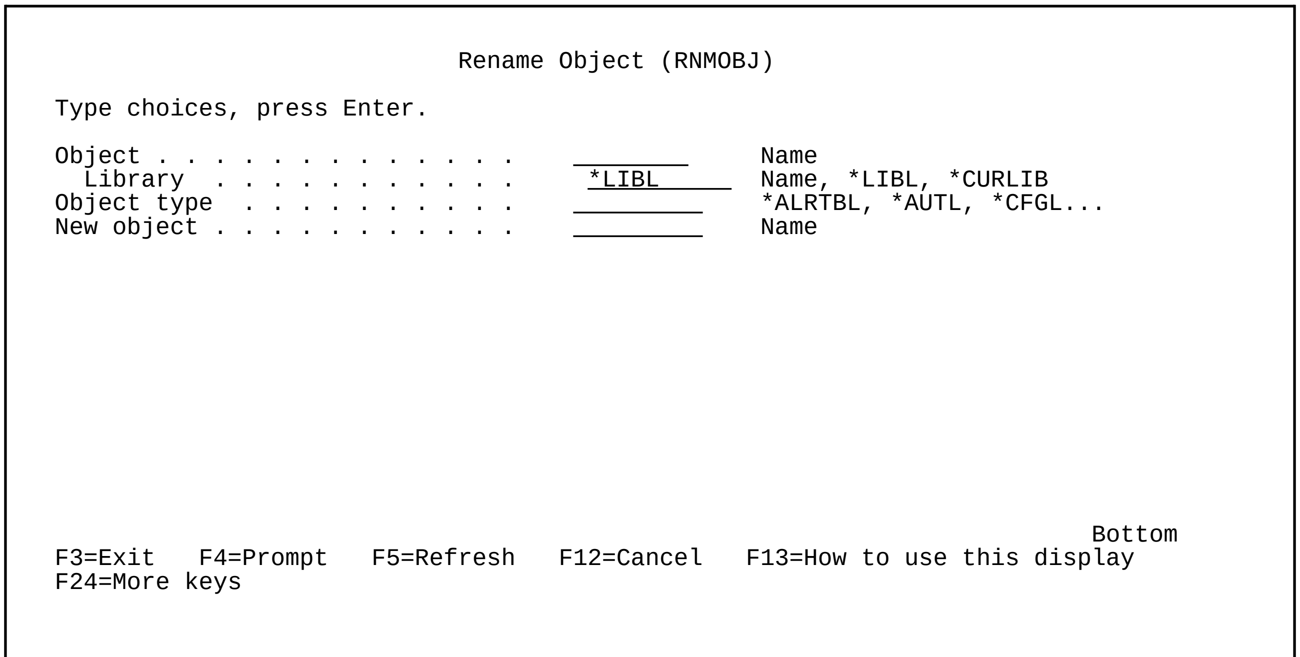
```
                               Copy Members

From file . . . . . . . :   QCSRC
  From library  . . . . :     JOEUSER

Type the file name and library name to receive the copied members.

  To file . . . . . . . .   QCSRC        Name, F4 for list
    To library  . . . . .     JOEUSER

To rename copied member, type New Name, press Enter.

Member         New Name
CTEST1         __________






                                                                    Bottom
F3=Exit             F4=Prompt     F5=Refresh     F12=Cancel
F19=Submit to batch
```

**Figure 14.10 The Program Development Manager's "Copy Members" display.**

### 14.1.5 Renaming Files

Instead of a command for renaming files, OS/400 has a command for naming objects: `RNMOBJ`. This makes things easier, because you use the same command to rename files, programs, and libraries.

If you enter `RNMOBJ` by itself and press F4, you get the command prompt display shown in Figure 14.11. Enter the object's name in the `Object` field, its library in the `Library` field (if you leave the default value of `*LIBL`, OS/400 will search the library list for the appropriate one) and the object's type in the field with that name. (As with copying, you need to specify this because you can have two objects with the same name in the same library, as long as they have two different object types.) In the `New object` field, enter the new name for the object.



```
                            Rename Object (RNMOBJ)

 Type choices, press Enter.

 Object . . . . . . . . . . . . .    ________      Name
   Library  . . . . . . . . . . .     *LIBL_____   Name, *LIBL, *CURLIB
 Object type  . . . . . . . . . .    _________     *ALRTBL, *AUTL, *CFGL...
 New object . . . . . . . . . . .    _________     Name

                                                                      Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

**Figure 14.11 Rename Object command prompt display.**

To rename a file called `SALES93` to `SALES93A`, enter `SALES93` in the `Object` field, `*FILE` in the `Object type` field, and `SALES93A` in the `New object` field. Assuming that `SALES93` was in a library in your library list, you could leave the `Library` field at its default value of `*LIBL`.

To do the same thing from the command line, enter this command:

```
RNMOBJ OBJ(SALES93) OBJTYPE(*FILE) NEWOBJ(SALES93A)
```

After you rename a file with either the command prompt display or the command line, OS/400 displays a message confirming the rename:

```
 Object SALES93 in JOEUSER type *FILE renamed SALES93A.
```

To rename a program, use the same syntax—just be sure to put `*PGM` as the object type:

```
RNMOBJ OBJ(RPGTEST) OBJTYPE(*PGM) NEWOBJ(EMPRPT)
```

Renaming a library would be similar, except that you enter an object type of `*LIB`.

#### 14.1.5.1 Renaming Members

Use the `RNMM` command to rename a member within a file. It needs to know three things:

| | |
|---|---|
| `FILE()` | The file where the member can be found. |
| `MBR()` | The name of the file to rename. |
| `NEWMBR()` | The member's new name. |

In the following example, Joe User renames the `CTEST1` member of his `QCSRC` file with a new name of `CTEST1A`:

```
RNMM FILE(QCSRC) MBR(CTEST1) NEWMBR(CTEST1A)
```

Because the command needs such simple, obvious information, it's very easy to enter it with positional parameters instead. The following command works just as well:

```
RNMM QCSRC CTEST1 CTEST1A
```

#### 14.1.5.2 Renaming Members with the Program Development Manager

Once you've listed a file's members with the Program Development Manager, you can rename one by entering the number 7 next to that member's name in the `Opt` column. The Program Development Manager will display the "Rename Members" screen, which has only one field for you to fill out: the new name of the member.

### 14.1.6 Deleting Files

Delete files with the `DLTF` command. Delete the file `CTEST1A` by entering

```
DLTF FILE(SALES93A)
```

or by entering the positional version:

```
DLTF SALES93A
```

Because programs are a different object type from files, there is a different command to delete them: `DLTPGM`. It uses the same syntax as `DLTF`. To delete a program called `CTEST1`, enter this:

```
DLTPGM CTEST1
```

#### 14.1.6.1 Deleting Members

Use the `RMVM` command to delete a member from a file. The `RMVM` command needs to know the name of the file and the name of the member within that file to delete. To delete the `CTEST1A` member from the `QCSRC` file, enter

```
RMVM FILE(QCSRC) MBR(CTEST1A)
```

or the positional version:

```
RMVM QCSRC CTEST1A
```

#### 14.1.6.2 Deleting Members with the Program Development Manager

Once you've listed a file's members with the Program Development Manager, you can delete one of these members by entering the number 4 next to that member's name in the `Opt` column.

The Program Development Manager will display the "Delete Members" screen. This has no fields to fill out; it shows the name of the member that you want to delete and a message at the top that tells you to "Press Enter to confirm your choices for Delete." (The plural "choices" is used because you could have entered the number 4 next to more than one member.) To abort the deletion, press F12 to return to the "Work with Members Using PDM" list of the file's members.

### 14.1.7 Editing Your Library List

Use the Edit Library List (`EDTLIBL`) command to do just what it says. (The Change Library List command—`CHGLIBL`—completely replaces the library list with a new one, as opposed to merely making edits. That's for advanced users, and is particularly useful in CL programs where altering the library list needs to be automated.)

The Display Library List command (`DSPLIBL`) displays a list of the libraries in your member list, if you just want to see what your library list looks like. (Don't confuse this with `DSPLIB`, the "Display Library" command, which lists the objects in a library.) Although `DSPLIBL` is not really an essential command, taking a look at its output makes it easier to understand some important things about library lists. Figure 14.12 shows the result when Joe User enters the `DSPLIB` command, which takes no parameters.

```
                              Display Library List
                                                             System:   NEPAS4


  Type options, press Enter.
    5=Display objects in library

  Opt  Library     Type      Text
       QSYS        SYS       System Library
       QSYS2       SYS       System Library for CPI's
       QUSRSYS     SYS       *IN USE
       QHLPSYS     SYS
       JOEUSER     CUR       Joe User's library
       URBASE      USR       UpRiteBase system library
       QTEMP       USR
       QGPL        USR       GENERAL PURPOSE LIBRARY
       USERTOOLS   USR
       QUSRTOOL    USR



                                                                       Bottom
  F3=Exit   F12=Cancel   F17=Top   F18=Bottom
  (C) COPYRIGHT IBM CORP. 1980, 1992.
```

**Figure 14.12 DSPLIBL output for Joe User.**

When Joe enters `EDTLIBL`, he will see a list of libraries, but he won't see as many as he sees when he enters `DSPLIBL`. That's because a complete library list has several parts, and `EDTLIBL` only lets him edit one part: the user part. Figure 14.13 shows what happens when he enters `EDTLIBL`.

```
                             Edit Library List

  Type new/changed information, press Enter.
    To add a library, type name and desired sequence number.
    To remove a library, space over library name.
    To change position of a library, type new sequence number.


  Sequence                   Sequence                  Sequence
   Number    Library          Number    Library         Number    Library
    010                        120                       230
    020      URBASE            130                       240
    030      QTEMP             140                       250
    040      QGPL              150
    050      USERTOOLS         160
    060      QUSRTOOL          170
    070                        180
    080                        190
    090                        200
    100                        210
    110                        220

  F3=Exit            F5=Refresh             F12=Cancel
```

**Figure 14.13 Edit Library List (EDTLIBL) screen.**

The user part of the library list, which comes at the end of the complete library list, is the part that you can control. You can reorder, delete, and add to its list of libraries.

As it shows in Figure 14.13, you manipulate the libraries and their order in the list by entering their names in the `Library` column and entering new numbers in the `Sequence Number` column. Each time you press Enter, the Edit Library List program makes (or tries to make) the changes that you indicated and reorders the numbers. The first library name on the list is always next to the number 20; you use the number 10 to indicate a library name that you want to move to the beginning of the list. For example, Figure 14.14 shows that Joe entered a "10" over the "40" next to the `QGPL` library's name.

```
                             Edit Library List

Type new/changed information, press Enter.
  To add a library, type name and desired sequence number.
  To remove a library, space over library name.
  To change position of a library, type new sequence number.


Sequence                     Sequence                  Sequence
 Number    Library            Number    Library         Number    Library
  010                          120                       230
  020      URBASE              130                       240
  030      QTEMP               140                       250
  010      QGPL                150
  050      USERTOOLS           160
  060      QUSRTOOL            170
  070                          180
  080                          190
  090                          200
  100                          210
  110                          220

F3=Exit            F5=Refresh             F12=Cancel
```

**Figure 14.14 Entering the new sequence number to move QGPL to the top of the library list.**

Figure 14.15 shows that after he presses Enter, `EDTLBL` moves `QGPL` to the beginning of the list, ahead of the `URBASE` library. The list has been renumbered, and `QGPL` becomes the new line 20, now that it's first on the list.

```
                            Edit Library List

 Type new/changed information, press Enter.
   To add a library, type name and desired sequence number.
   To remove a library, space over library name.
   To change position of a library, type new sequence number.


 Sequence                  Sequence                  Sequence
  Number    Library         Number    Library         Number    Library
   010                       120                       230
   020      QGPL             130                       240
   030      URBASE           140                       250
   040      QTEMP            150
   050      USERTOOLS        160
   060      QUSRTOOL         170
   070                       180
   080                       190
   090                       200
   100                       210
   110                       220

 F3=Exit            F5=Refresh            F12=Cancel
```

**Figure 14.15 The effect of pressing Enter after entering a 10 next to the QGPL library to move it to the front of the list.**

To move a library name between two others, enter a number in the sequence column next to it that falls between the two numbers at its destination. For example, to move the `URBASE` library name between the `USERTOOLS` and `QUSRTOOL` libraries, you could enter the number 55 (or any other number between 50 and 60) next to it and press Enter.

To delete a library from the library list, type over its name with the space bar and press Enter.

To insert a new library:

1. Enter its name in any blank space in the `Library` column.

2. Give it a number showing where you want it to be placed in the list.

3. Press Enter.

After you make any successful change to your library list, you'll see the message "Library list changed" at the bottom of your screen.

A typical unsuccessful change would be the attempted addition of a non-existent library. If you tried to add the non-existent library `POTRZEBIE` to your list, you get the message "Library POTRZEBIE not found."

Any changes you make to your library list will only remain in effect for the current OS/400 session. For help in making permanent changes to your library list, see your system administrator.

#### 14.1.7.1 Changing Your Current Library

There are two ways to reset your current library:

- As we saw in section 13.1, "Starting Up," you can enter a specific library in the "Current library" field of the signon screen when you enter you user ID and password.

- You can use the `CHGCURLIB` command. This command's command prompt display shows that it only needs one parameter: the name of the library to make current. To change your current library to one called `URBASE` from the command line, enter

  ```
  CHGCURLIB CURLIB(URBASE)
  ```

  or even just this:

  ```
  CHGCURLIB URBASE
  ```

### 14.1.8 Creating and Deleting Libraries

There's a very good chance that the first time you sign on, you will find that the system administrator has created a personal library named after your user ID and that you do not have the privileges to create or delete other libraries. In this situation, you keep your files in your own library and leave the creation and deletion of libraries to the system administrator.

On the other hand, you may be told "Here's your user ID name. The first thing you have to do when you sign on is to create a library for yourself." The Create Library (`CRTLIB`) command makes it very easy. To create a library called `MYLIB`, enter the following:

```
CRTLIB LIB(MYLIB)
```

Because libraries are objects, just as files are, you follow the same rules for making up library names that you do for file names. (For more information, see section 13.2, "File Names.") The one extra rule—a guideline, really, because the system won't prevent you from doing this—is to avoid beginning a library name with the letter "Q," because this indicates a system library, and could give other users the wrong idea.

If the command worked, you will see the message "Library MYLIB created" at the bottom of your screen.

To delete a library, use the Delete Library (`DLTLIB`) command:

```
DLTLIB LIB(MYLIB)
```

If successful, you will see the message "Library MYLIB deleted" at the bottom of your screen.

Because both of these commands have only one required parameter, there is no problem with entering that parameter without its parameter name. Entering

```
CRTLIB MYLIB
```

or

```
DLTLIB MYLIB
```

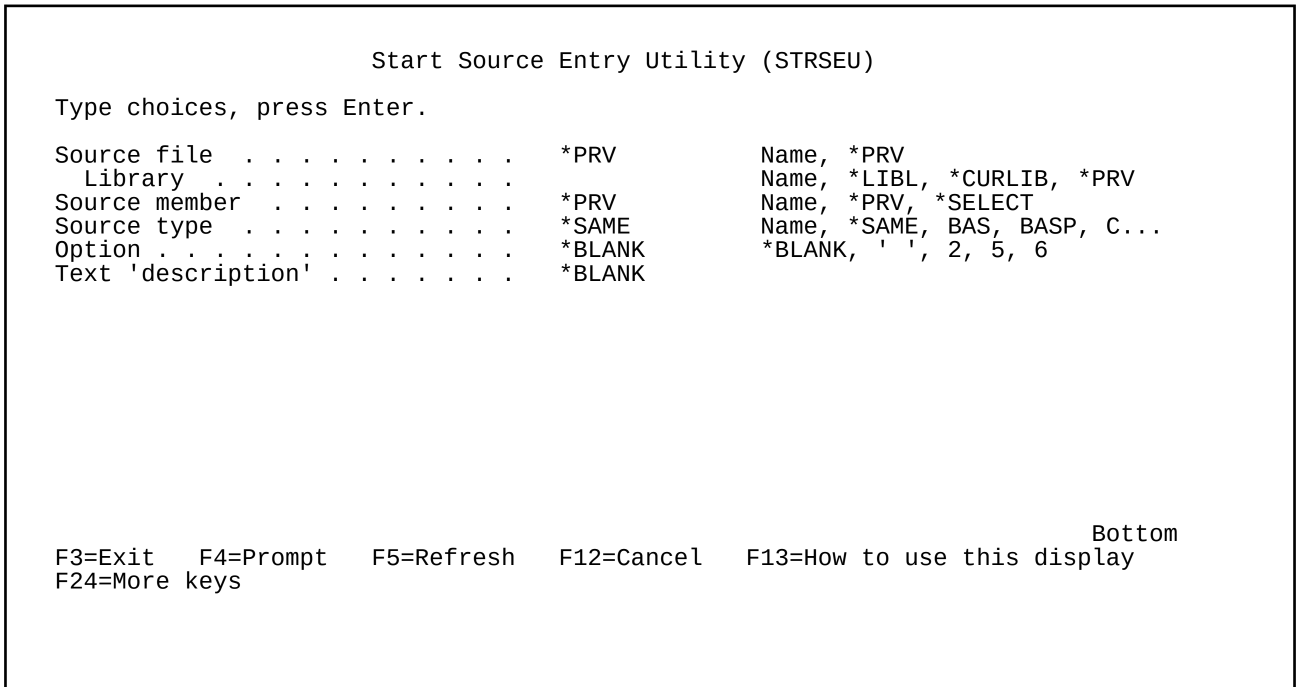
will both work.

# Chapter 15 The OS/400 SEU Text Editor

As we saw in section 13.3.1, "Physical, Source Physical, and Logical Files," the most popular use of the OS/400 Source Entry Utility (SEU) text editor—in fact, the reason for its name—is for creating and editing source code. However, you don't have to limit your use of SEU to this; you can use it to create any text file you like.

## 15.1 Entering SEU

To enter SEU, use the Start SEU command (`STRSEU`). If you enter it with no parameters and press F4, you'll see its command prompt display screen, as shown in Figure 15.1.



```
                      Start Source Entry Utility (STRSEU)

 Type choices, press Enter.

 Source file  . . . . . . . . . .   *PRV          Name, *PRV
   Library  . . . . . . . . . . .                 Name, *LIBL, *CURLIB, *PRV
 Source member  . . . . . . . . .   *PRV          Name, *PRV, *SELECT
 Source type  . . . . . . . . . .   *SAME         Name, *SAME, BAS, BASP, C...
 Option . . . . . . . . . . . . .   *BLANK        *BLANK, ' ', 2, 5, 6
 Text 'description' . . . . . . .   *BLANK









                                                                       Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

**Figure 15.1 The STRSEU text editor's command prompt display screen.**

You need to tell `STRSEU` the file with the member that you want to edit and the name of the member. Note how the default values for both of these are `*PRV`; these mean "previous value." In other words, if you had entered `STRSEU` with no parameters and pressed Enter, it would have brought up the file member that you had most recently edited.

`STRSEU` will not create a new file for you, but it will create a new member in an existing file. In addition to entering the member's name in the "Source Member" field, enter its type in the "Source Type" field. The default source type is TXT. (Even if the member exists, nothing prevents you from entering a new value in the "Source Type" field. In fact, this is the easiest way to change a file member's type.)

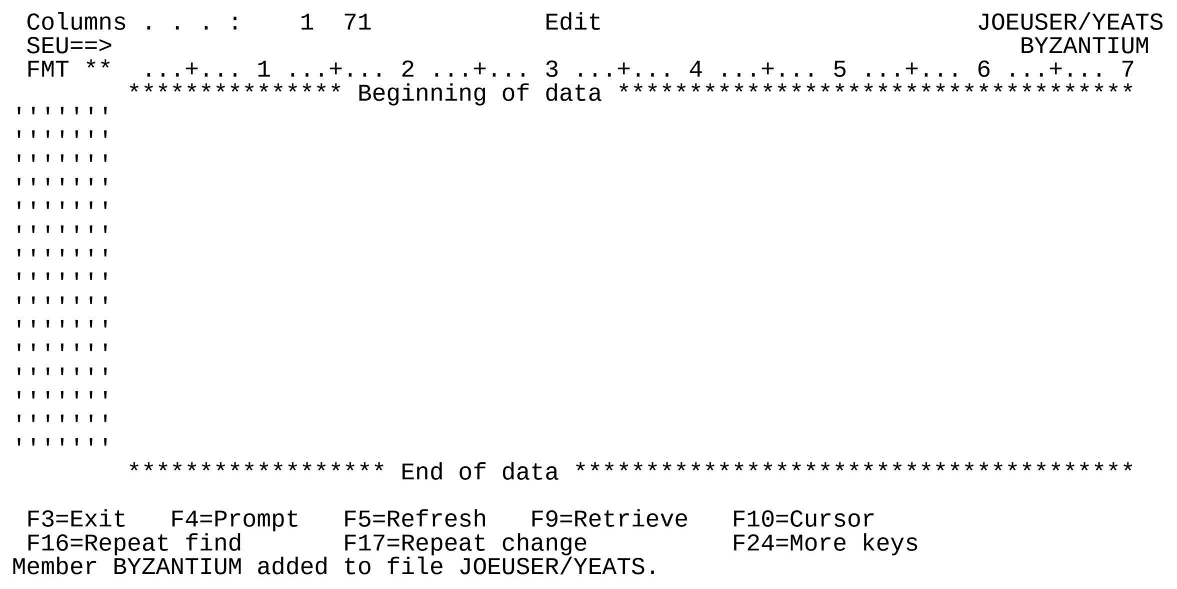
Of course, you could skip the command prompt display screen by entering the file and member name at the command line. For example, let's say you have a file called `YEATS` and you want to edit a member called `BYZANTIUM`. If you include the parameter names, you would enter the following:

```
STRSEU SRCFILE(YEATS) SRCMBR(BYZANTIUM)
```

If you wanted to leave out the parameter names and enter the file and member names as positional parameters, you would enter this:

```
STRSEU YEATS BYZANTIUM
```

SEU starts up, and if the member exists, it shows the beginning of the member. Otherwise, it shows a blank editing area and a message at the bottom informing you that the member has been added to the file. Figure 15.2 shows an example of this.



```
  Columns . . . :    1  71           Edit                          JOEUSER/YEATS
  SEU==>                                                              BYZANTIUM
  FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
         *************** Beginning of data ************************************
 '''''''
 '''''''
 '''''''
 '''''''
 '''''''
 '''''''
 '''''''
 '''''''
 '''''''
 '''''''
 '''''''
 '''''''
 '''''''
 '''''''
 '''''''
         ****************** End of data ***************************************

  F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F10=Cursor
  F16=Repeat find       F17=Repeat change          F24=More keys
 Member BYZANTIUM added to file JOEUSER/YEATS.
```

**Figure 15.2 Opening SEU screen for a new member.**

The `SEU==>` prompt points to the SEU command line, where you can enter certain SEU commands. The F10 key moves the cursor back and forth between the SEU command line and the data area.

The column of apostrophes (`'''''''`) down the left shows the location of the sequence number field for each line. This will show numbers next to a file member's lines once they exist; before then, the apostrophes show places where you can enter new lines of text. This is where you enter the commands known as *line commands,* which you use to delete, copy, and move lines of your file member.

The `Columns: 1 71` message in the upper-left show that you are looking at columns 1 through 71 of the file member named in the upper-right. SEU can edit file members that are too wide for the display and has commands to scroll to the left or right to look at different parts of it. In these situations, the `Columns` part of the screen is pretty handy.

### 15.1.1 Entering SEU from the Program Development Manager

When you've listed a file's members with the Program Development Manager, you can edit one of these members by entering the number 2 next to that member's name in the `Opt` column. The Program Development Manager will start up SEU with that member for you to edit.

## 15.2 Line Commands

In addition to command-line commands entered at the `SEU==>` prompt, many SEU operations are performed with line commands entered in the sequence number column. You can add, delete, copy, and move lines by entering one- or two-character commands in this column and pressing the Enter key. (This will be familiar to users of the MVS ISPF editor, and CMS users who detect a similarity to the XEDIT text editor will find that line commands are SEU's counterpart to XEDIT line commands.)

Line commands are not limited to the sequence number column of the contents of your file; you can also enter them in the sequence number column of the `*** Beginning of data ***` and `*** End of data ***` lines.

You can enter a line command anywhere on the sequence number. For example, you can enter the command to delete two lines (`D2`) on the line numbered "0001.00" like this

```
D201.00
```

or

```
00D2.00
```

This can obviously lead to confusion if you enter line commands that use numbers (like `D2`) on a sequence number column that is displayed as numbers instead of as apostrophes (like "0001.00") so be careful.

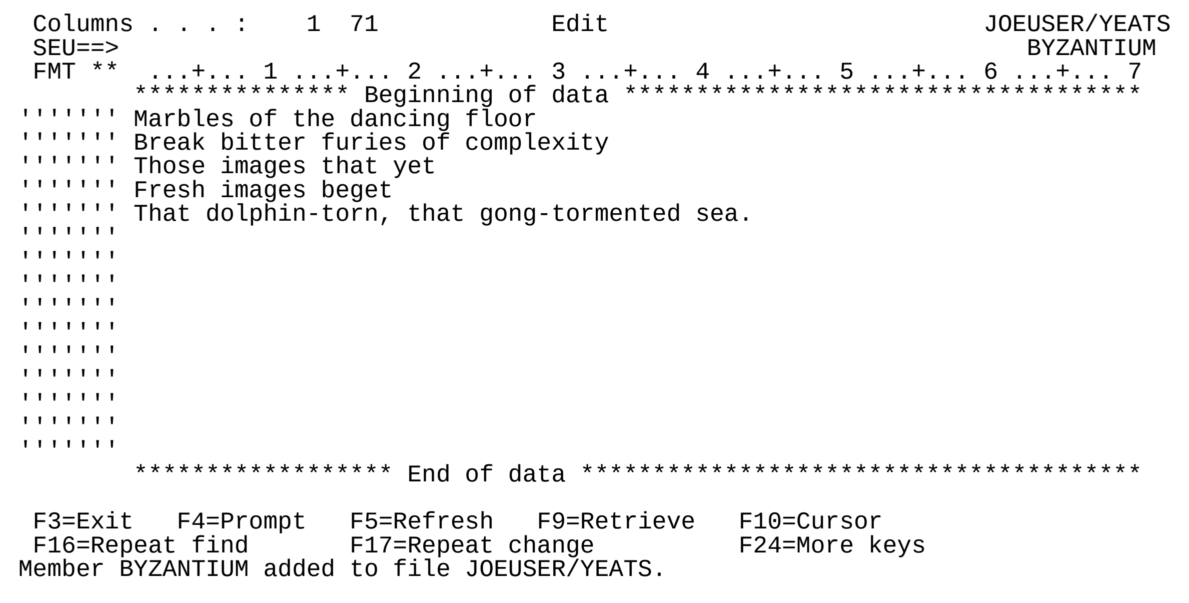
If you ever enter a line command and then realize, before you press Enter, that you didn't mean to enter it, just type spaces over it. The next time you press Enter, SEU restores the numbers that were there before you entered the line command.

## 15.2.1 Adding New Lines

We saw that apostrophes next to a line on the display mean that you can enter new lines of text there. Since new lines are automatically numbered when you press Enter, all existing lines will have numbers to the left of them. You can type over the text of existing lines as easily as you can enter new text on blank lines.

When you press Enter, SEU removes any blank lines that still have apostrophes in the sequence number field and that are below the last newly entered line, and the `End of data` message jumps up to just below the last real line of the file member.

Figure 15.3 shows the same screen as Figure 15.2 with several lines of text that Joe User has added. He has not pressed Enter yet.



```
 Columns . . . :    1  71           Edit                          JOEUSER/YEATS
 SEU==>                                                                BYZANTIUM
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
        *************** Beginning of data ************************************
''''''' Marbles of the dancing floor
''''''' Break bitter furies of complexity
''''''' Those images that yet
''''''' Fresh images beget
''''''' That dolphin-torn, that gong-tormented sea.
'''''''
'''''''
'''''''
'''''''
'''''''
'''''''
'''''''
'''''''
'''''''
'''''''
        ****************** End of data ***************************************

 F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F10=Cursor
 F16=Repeat find       F17=Repeat change          F24=More keys
Member BYZANTIUM added to file JOEUSER/YEATS.
```

**Figure 15.3 Empty SEU screen with some text entered, before Enter is pressed.**

Figure 15.4 shows the same screen after Enter is pressed. Notice how the entered lines have been numbered, and the blank lines are gone.

```
  Columns . . . :    1  71           Edit                           JOEUSER/YEATS
  SEU==>                                                               BYZANTIUM
  FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
         *************** Beginning of data ************************************
 0001.00 Marbles of the dancing floor
 0002.00 Break bitter furies of complexity
 0003.00 Those images that yet
 0004.00 Fresh images beget
 0005.00 That dolphin-torn, that gong-tormented sea.
         ****************** End of data ***************************************








  F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F10=Cursor
  F16=Repeat find       F17=Repeat change          F24=More keys
```

**Figure 15.4 Effect of pressing Enter when viewing the screen shown in Figure 15.3.**

You can easily insert blank lines for new lines of text with the `I` line command. If you type an `I` over the number in a line's sequence number field, the next time you press Enter, a new blank line will be inserted after that line. (To insert new lines before the current first line, enter the `I` command in the same position on the `Beginning of data` line.)

If you enter a number after the `I`, that many new lines will be inserted. For example, in Figure 15.5, Joe User has entered `I3` to the left of line 2, but he has not pressed Enter yet. Figure 15.6 shows the effect of pressing Enter to insert those three lines.

```
  FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
         *************** Beginning of data ************************************
 0001.00 Marbles of the dancing floor
 I302.00 Break bitter furies of complexity
 0003.00 Those images that yet
 0004.00 Fresh images beget
 0005.00 That dolphin-torn, that gong-tormented sea.
         ****************** End of data ***************************************
```

**Figure 15.5 Entering the line command to insert blank lines after line 2.**

In Figure 15.6, new lines have not really been added yet; note that line 2 is still line 2 and line 3 is still line 3. You have 3 potential new lines that serve the same purpose as the blank lines that you saw when you first started up SEU with a new file member.

```
  FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
        *************** Beginning of data ************************************
 0001.00 Marbles of the dancing floor
 0002.00 Break bitter furies of complexity
 '''''''
 '''''''
 '''''''
 0003.00 Those images that yet
 0004.00 Fresh images beget
 0005.00 That dolphin-torn, that gong-tormented sea.
        ****************** End of data ****************************************
```

**Figure 15.6 The effect of pressing Enter after entering the command shown in Figure 15.5.**

If you enter text on only one of these new lines, as shown in Figure 15.7, and then press Enter, you'll get the result shown in Figure 15.8: the line with new text and the blank line before it are assigned line numbers and the blank line after it is removed.

```
  FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
        *************** Beginning of data ************************************
 0001.00 Marbles of the dancing floor
 0002.00 Break bitter furies of complexity
 '''''''
 ''''''' Bitter furies of complexity!  I'll say!  That crazy Yeats!
 '''''''
 0003.00 Those images that yet
 0004.00 Fresh images beget
 0005.00 That dolphin-torn, that gong-tormented sea.
        ****************** End of data ****************************************
```

**Figure 15.7 Entering text on only one of the new lines inserted in Figure 15.6.**

```
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
        *************** Beginning of data ************************************
0001.00 Marbles of the dancing floor
0002.00 Break bitter furies of complexity
0002.01
0002.02 Bitter furies of complexity!  I'll say!  That crazy Yeats!
0003.00 Those images that yet
0004.00 Fresh images beget
0005.00 That dolphin-torn, that gong-tormented sea.
        ****************** End of data ***************************************
```

**Figure 15.8 The effect of pressing Enter after entering the new line shown in Figure 15.7.**

Note how the new line numbers are decimal numbers between 2.00 and 3.00. When you save the file member, SEU offers you the choice of renumbering the lines with whole numbers or leaving them as they are. Renumbering them with whole numbers is the default.

If you press Enter after adding text to the last of the new lines that are waiting for text, SEU inserts a new blank line under it and positions your cursor at its beginning. This is handy for entering many lines of text, because you can just type a line, press Enter, type another line, press Enter, and repeat this until you have entered all of your text.

### 15.2.2 Moving Your Cursor Around

Your up, down, left, and right cursor keys move the cursor in the direction in which they point.

The Tab key helps you move around more quickly. To move your cursor to the beginning of the previous line, use the Backtab key (with a PC that is emulating a 3270 terminal, press the Shift key and the Tab key simultaneously). If your cursor is on a line of text, Tab or Backtab move your cursor to the sequence number column. If your cursor is on the sequence number column, pressing either key jumps your cursor to the beginning of the appropriate line.

## 15.3 Inserting, Deleting, and Typing over Words and Characters

To add text to blank lines, just move your cursor to the line and type. When you need to move your cursor back to the command line, press F10.

To delete an individual character, move your cursor there and press your Delete key. On a 3270 terminal, the delete key has a lower case "a" with a proofreader's symbol for deletion: a line through it that forms a loop. When emulating a 3270 terminal, your emulation software probably has your PC's Delete key doing this job.

To type over existing text, just move your cursor where you want the new text and type.

To insert text, move your cursor to the place where you want to insert it and press the Insert key. On a 3270 terminal, this key has the letter "a" with a carat symbol (^) over it. When you press it, a carat symbol should appear at the bottom of your screen to indicate that you are in insert mode. (When emulating a 3270, your cursor may change shape.) Text that you type in moves any text currently to the right of the cursor further to the right.

To return to overstrike mode while using a 3270 terminal, press the key marked "Reset." The carat symbol should disappear, and newly typed text takes the place of the characters at the cursor. (When your keyboard "locks up," or refuses to accept input, the Reset key is also useful for freeing up the keyboard.) On most PCs emulating a 3270, the Insert key puts you in insert mode and the Escape key stands in for the Reset key.

## 15.3.1 Duplicating Lines

Think of it as "repeating" lines, it will be easier to remember the line command: `RP`. Enter `RP` by itself and press Enter to make a single copy of a line. For example, if you enter `RP` on the third line in the text shown in Figure 15.9, and then press Enter, you get the result shown in Figure 15.10.

```
        *************** Beginning of data ************************************
0001.00 Marbles of the dancing floor
00RP.00 Break bitter furies of complexity
0003.00 Those images that yet
0004.00 Fresh images beget
0005.00 That dolphin-torn, that gong-tormented sea.
        ****************** End of data ***************************************
```

**Figure 15.9 Entering the command to repeat a line.**

```
        *************** Beginning of data ************************************
 0001.00 Marbles of the dancing floor
 0002.00 Break bitter furies of complexity
 0002.01 Break bitter furies of complexity
 0003.00 Those images that yet
 0004.00 Fresh images beget
 0005.00 That dolphin-torn, that gong-tormented sea.
        ****************** End of data ****************************************
```

**Figure 15.10 Effect of pressing Enter after entering RP command in Figure 15.9.**

Entering a number after the `RP` command tells SEU to repeat the line that many times. For example, if `RP3` had been entered on line 2 of Figure 15.9 instead of `RP`, the result would have looked like Figure 15.11.

```
        *************** Beginning of data ************************************
 0001.00 Marbles of the dancing floor
 0002.00 Break bitter furies of complexity
 0002.01 Break bitter furies of complexity
 0002.02 Break bitter furies of complexity
 0002.03 Break bitter furies of complexity
 0003.00 Those images that yet
 0004.00 Fresh images beget
 0005.00 That dolphin-torn, that gong-tormented sea.
        ****************** End of data ****************************************
```

**Figure 15.11 Effect of pressing Enter if RP3 had been entered at line 2 in Figure 15.9.**

## 15.3.2 Deleting Lines

The letter `D` in a line's sequence number field deletes that line the next time you press Enter. A number immediately following it tells SEU to delete that many lines, starting with the line where the command is entered. In Figure 15.12, two lines are about to be deleted.

```
        *************** Beginning of data ************************************
```

```
 0001.00 Marbles of the dancing floor
 000D200 Break bitter furies of complexity
 0003.00 Those images that yet
 0004.00 Fresh images beget
 0005.00 That dolphin-torn, that gong-tormented sea.
         ****************** End of data ****************************************
```

**Figure 15.12 Entering the 2D line command to delete two lines in SEU.**

After you press Enter, they're gone. Figure 15.13 shows the result; note how the lines have not been renumbered.

```
         *************** Beginning of data ************************************
 0001.00 Marbles of the dancing floor
 0004.00 Fresh images beget
 0005.00 That dolphin-torn, that gong-tormented sea.
         ****************** End of data *****************************************
```

**Figure 15.13 Result of the 2D line command entered in Figure 15.12.**

Put the letter `D` twice in a sequence number field without any number to indicate that the line begins or ends a block that you want to delete. If you press Enter while only one line has the `DD`, SEU leaves it there until it has a partner, and displays the message "Block command not complete" as a reminder. In Figure 15.14, SEU is ready to delete all but the first and last lines of text the next time you press Enter.

```
         *************** Beginning of data ************************************
 0001.00 Marbles of the dancing floor
 DD02.00 Break bitter furies of complexity
 0003.00 Those images that yet
 00DD.00 Fresh images beget
 0005.00 That dolphin-torn, that gong-tormented sea.
         ****************** End of data ****************************************
```

**Figure 15.14 Entering the DD line command to delete a block of lines in SEU.**

After you press Enter, the lines with the `DD` commands and all the lines between them are removed.

This command is particularly useful when the beginning and end of the block that you want to delete are not on the same screen, because the alternative (counting the number of lines so that you can put a number after a single `D`) is a lot of trouble.

### 15.3.3 Copying Lines

Copying is similar to deletion except that you use the letter `C` to indicate the line or lines to copy and you must indicate a destination for the copied text. SEU gives you three options for indicating the text to copy:

- Enter a single `C` in a line's sequence number field if you only want to copy that one line.
- Enter a single `C` followed by a number to indicate how many lines to copy.
- Enter `CC` at the first and last lines of the block to copy.

In addition to indicating the line or lines to copy, you must indicate where to copy them. Two line commands make this possible:

| | |
|---|---|
| B | When Enter is pressed, copy the block to the line before the line with this command. |
| A | When Enter is pressed, copy the block to the line after the line with this command. |

In Figure 15.15, the third, fourth, and fifth lines are about to get copied above the first line, to the beginning of the file member.

```
        *************** Beginning of data *************************************
0000B.00 Marbles of the dancing floor
0002.00 Break bitter furies of complexity
CC03.00 Those images that yet
0004.00 Fresh images beget
0CC5.00 That dolphin-torn, that gong-tormented sea.
        ****************** End of data ****************************************
```

**Figure 15.15 Using the CC and B line commands to copy a block in SEU.**

Figure 15.16 shows how it looks after you press Enter.

```
        *************** Beginning of data ************************************
0000.01 Those images that yet
0000.02 Fresh images beget
0000.03 That dolphin-torn, that gong-tormented sea.
0001.00 Marbles of the dancing floor
0002.00 Break bitter furies of complexity
0003.00 Those images that yet
0004.00 Fresh images beget
0005.00 That dolphin-torn, that gong-tormented sea.
        ****************** End of data ***************************************
```

**Figure 15.16 Result of the CC and B line commands entered in Figure 15.15.**

## 15.3.4 Moving Lines

Moving is similar to copying, except that after you press Enter, the original lines are no longer there—they're moved to their new location. As with copying, there are three ways to specify the block to move, but these use the letter `M`:

- Enter a single `M` in a line's sequence number field if you only want to move that one line.
- Enter a single `M` followed by a number to indicate how many lines to move.
- Enter `MM` at the first and last lines of the block to move.

To specify the destination of the block to move, use the letters `B` or `A` the same way you do to specify the destination of a block to copy.

## 15.4 Searching for Text

To search for text, use the `FIND` command from the SEU command line. (Use the F10 key to move your cursor to the command line.) If you like, you can abbreviate `FIND` to just `F`. If your search target has spaces in it, enclose it in apostrophes or quotation marks.

Figure 15.17 shows the command to search for the string "images" just before Enter is pressed.

```
 Columns . . . :    1  71           Edit                              JOEUSER/YEATS
 SEU==> find images                                                      BYZANTIUM
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
        *************** Beginning of data *************************************
0001.00 Marbles of the dancing floor
0002.00 Break bitter furies of complexity
0003.00 Those images that yet
0004.00 Fresh images beget
0005.00 That dolphin-torn, that gong-tormented sea.
        ****************** End of data ****************************************
```

**Figure 15.17 Entering the command to search for the string "images" on the SEU command line.**

After you press Enter, if the search was successful, the cursor jumps to the beginning of the found string and a message at the bottom of the screen informs you that the string was found. Figure 15.18 shows an example.

```
 Columns . . . :    1  71           Edit                              JOEUSER/YEATS
 SEU==>                                                                  BYZANTIUM
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
        *************** Beginning of data **************************************
0001.00 Marbles of the dancing floor
0002.00 Break bitter furies of complexity
0003.00 Those images that yet
0004.00 Fresh images beget
0005.00 That dolphin-torn, that gong-tormented sea.
        ****************** End of data *****************************************
```

```
  F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F10=Cursor
  F16=Repeat find      F17=Repeat change         F24=More keys
 String images found.
```

**Figure 15.18 Result of a successful search in SEU.**

If this search had been unsuccessful, the cursor would have remained on the command line and the status line at the bottom of the screen would have told you "String images not found."

As the bottom of the screen shows, pressing F16 searches for the next occurrence of the same string.

### 15.4.1 Case Sensitivity

The `FIND` command can do case-sensitive and case-insensitive searches. You control this with the `SET MATCH` command. To make the searches case-sensitive, enter

```
SET MATCH ON
```

at the SEU command line. All uses of the FIND command will then look for exact case matches. To set it to ignore case when searching, you can guess what the command is:

```
SET MATCH OFF
```

## 15.5 Saving Your Changes

To save your file member and then continue working, simply enter `SAVE` at the SEU command line. To save the edited file under a new name, enter `SAVE` followed by that name. For example, entering

```
SAVE OHYEAH
```

saves the file member with the name `OHYEAH`. If there is no such member in the file, SEU creates it and displays a message similar to the following at the bottom of the screen:

```
Member OHYEAH added to file JOEUSER/YEATS.
```

If a member by this name had already existed, SEU would have warned you with the following message:

```
Member OHYEAH already exists.  Press Enter to confirm.
```

If you want to abort this save, space over the SAVE command at the SEU command line before the next time you press Enter.

## 15.6 Quitting SEU

To quit SEU, press F3. SEU will display a screen similar to the one shown in Figure 15.19.

```
                                    Exit

  Type choices, press Enter.

    Change/create member  . . . . . . .   N            Y=Yes, N=No
      Member  . . . . . . . . . . . . .   BYZANTIUM    Name, F4 for list
      File  . . . . . . . . . . . . . .   YEATS        Name, F4 for list
        Library . . . . . . . . . . . .     JOEUSER    Name
      Text  . . . . . . . . . . . . . .

      Resequence member . . . . . . . .   Y            Y=Yes, N=No
        Start . . . . . . . . . . . . .   0001.00      0000.01-9999.99
        Increment . . . . . . . . . . .   01.00        00.01-99.99

    Print member  . . . . . . . . . . .   N            Y=Yes, N=No

    Return to editing . . . . . . . . .   N            Y=Yes, N=No

    Go to member list . . . . . . . . .   Y            Y=Yes, N=No



  F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel
```

**Figure 15.19 The "Exit SEU" screen.**

If you had any unsaved changes, the default value for "Change/create member" will be "Y" for "Yes."

The other fields in Figure 15.19 are self-explanatory. As with any entry display, you can use your Tab or cursor keys to move your cursor to any field and then type in a new value or press F1 to find out more about that field.

## 15.7 Other SEU Features

### 15.7.1 SEU On-line Help

On-line help in SEU follows the same conventions that it does elsewhere in OS/400:

- To learn more about function keys, move your cursor to the their descriptions at the bottom of the screen and press F1.
- To learn more about SEU commands, move your cursor to the sequence number field or the SEU command line and press F1.
- To learn more about a specific command, enter that command at the SEU command line and press F1 instead of Enter.

### 15.7.2 Syntax Prompting

SEU has built-in intelligence that can be a great help in writing programs in any language. Because it knows the extended attribute of the file member you are editing, it knows the programming language you are using to write your source code. This means that it can help you with that particular language and point out syntax errors as you write so that you don't have to wait for your compilation to bomb to see what's wrong with your source code.

This is particularly useful when writing CL programs. For more information, see section 16.2, "Command Files."

# Chapter 16 Using an OS/400 System

## 16.1 Printing Text Files

Many OS/400 programs offer printing as one of their features; section 16.1.1, "Printing a File Member from the Program Development Manager or SEU," describes two handy ways to print file members.

To print something from the OS/400 command line, we can use a command that we've already seen: `CPYSRCF` (Copy Source File). The following command uses the predefined value *PRINT to tell OS/400: "copy the BYZANTIUM member of the YEATS file, but not to another file member—instead, send it to the printer."

```
CPYSRCF FROMFILE(JOEUSER/YEATS) TOFILE(*PRINT) FROMMBR(BYZANTIUM)
```

Using positional parameters, the command would look like this:

```
CPYSRCF YEATS *PRINT BYZANTIUM
```

When you tell the system to print something, it doesn't really send it directly to the printer. Instead, it goes to a program called a print spooler, which acts as a traffic cop for the various print jobs as they come up.

Once you've sent something to a print spooler, that doesn't mean that it's going to be printed. A program known as a printer writer must take the print job from the spooler and pass it along to the printer.

This and several other useful tasks (like checking the print queue and cancelling print jobs, as you'll see in the following sections) can be done with the Work with Spooled Files (`WRKSPLF`) command, which displays the "Work with Printer Output" display. `WRKSPLF` doesn't need any parameters, although there are several optional ones that you can learn about by entering the command and pressing F4 instead of Enter. (Another way to display the "Work with Printer Output" screen is by selecting choice 1 from the Operational Assistant. Type `GO ASSIST` to start up the Operational Assistant.)

Figure 16.1 shows a sample "Work with Printer Output" display after Joe User entered the command above to print the `BYZANTIUM` member of the `YEATS` file.

```
                           Work with Printer Output
                                                             System:   NEPAS4
User . . . . . :   JOEUSER

Type options below, then press Enter.  To work with printers, press F22.
  2=Change   3=Hold   4=Delete   5=Display          6=Release   7=Message
  9=Work with printing status    10=Start printing   11=Restart printing

     Printer/
```

```
Opt    Output      Status
     Not Assigned
       QSYSPRT     Not assigned to printer (use Opt 10)









                                                                     Bottom
F1=Help      F3=Exit   F5=Refresh   F9=Command line   F11=Dates/pages/forms
F12=Cancel   F21=Select assistance level   F22=Work with printers
```

**Figure 16.1 WRKSPLF display for controlling printer output.**

Notice that it doesn't mention "YEATS" or "BYZANTIUM" anywhere. Instead, it shows the name assigned by the program that put it in the print spooler (in this case, `CPYSRCF`): `QSYSPRT`. To make sure that QSYSPRT is the print job that you think it is, you can display it by moving your cursor into the `Opt` column next to it and entering the number 5. This displays the text that is waiting to go to the printer on a screen similar to the one used by the Display Physical File Member (`DSPPFM`) command.

The column showing `QSYSPRT` is labeled "Printer/Output." It shows the names of the various printers available, and indented under each printer name, the jobs waiting to go there. Jobs that are not bound for any particular printer are indented under the name "Not Assigned."

Figure 16.1 shows that `QSYSPRT` is not headed for any printer. The top of the display shows that entering the number 10 next to `QSYSPRT` will "start printing" it. Figure 16.2 shows what happens when we do this to a printer output file that has not been assigned to a printer.

```
                          Assign Output to a Printer

  Printer output . . :   QSYSPRT

This printer output is not assigned to a printer.
To print the output, type the printer name below and then press Enter.

  Printer  . . . . . .  __________    Name, F4 for list
```

```
 F1=Help   F3=Exit   F12=Cancel
```

**Figure 16.2 Assigning printer output to a particular printer.**

As the "Assign Output to a Printer" screen shows, pressing F4 displays a list of the valid printers that are connected to your AS/400, allowing you to pick one as the destination for your print job. If you don't know the name assigned to the printer where you want to send your output, ask your system administrator.

Figure 16.3 shows how the "Work with Printer Output" screen looks once the print job is in progress. The printer assigned in Figure 16.2 was called NEPPRT1.

```
                            Work with Printer Output
                                                             System:   NEPAS4
 User . . . . . :   JOEUSER

 Type options below, then press Enter.  To work with printers, press F22.
   2=Change   3=Hold   4=Delete   5=Display          6=Release   7=Message
   9=Work with printing status    10=Start printing   11=Restart printing

      Printer/
 Opt    Output      Status
      NEPPRT1
        QPSUPRTF    Printing starting or ending (use F5)







                                                                       Bottom
 F1=Help      F3=Exit   F5=Refresh   F9=Command line   F11=Dates/pages/forms
 F12=Cancel   F21=Select assistance level   F22=Work with printers
```

**Figure 16.3 The print job on its way to the printer.**

The `Status` column shows the status of the print job at the instant the screen is displayed. To update the display press F5, the "Refresh" key. Other typical status messages are "Waiting to print" for a job that has been assigned to a printer but hasn't reached its turn yet and "Printing page 1 of 1" (with whatever appropriate numbers) for a job currently being printed.

If your printer output doesn't automatically get sent to a printer, it would be pretty annoying to have to go through all of these steps every time you print something. Your user profile stores the name of your default printer destination, and you can change your profile with the `CHGPRF` command with the following steps:

1. Enter `CHGPRF` and press F4 to see the CHGPRF parameters. It will only show you a few of the parameters, and "Print Device" won't be one of them.

2. The bottom of this screen will tell you that pressing F10 shows you "Additional Parameters." Press F10, and you will see additional parameters, but you still won't see "Print Device."

3. You will see the `More...` message in the lower right, and after pressing Page Down (or Scroll Up) you will see the "Print Device" field. Enter the name of the printer you learned about from the "Assign Output to a Printer" screen (or from your system administrator) here and all of your printed output will then be sent to that printer.

4. Press Enter to show that you are done editing the `CHGPRF` parameters.

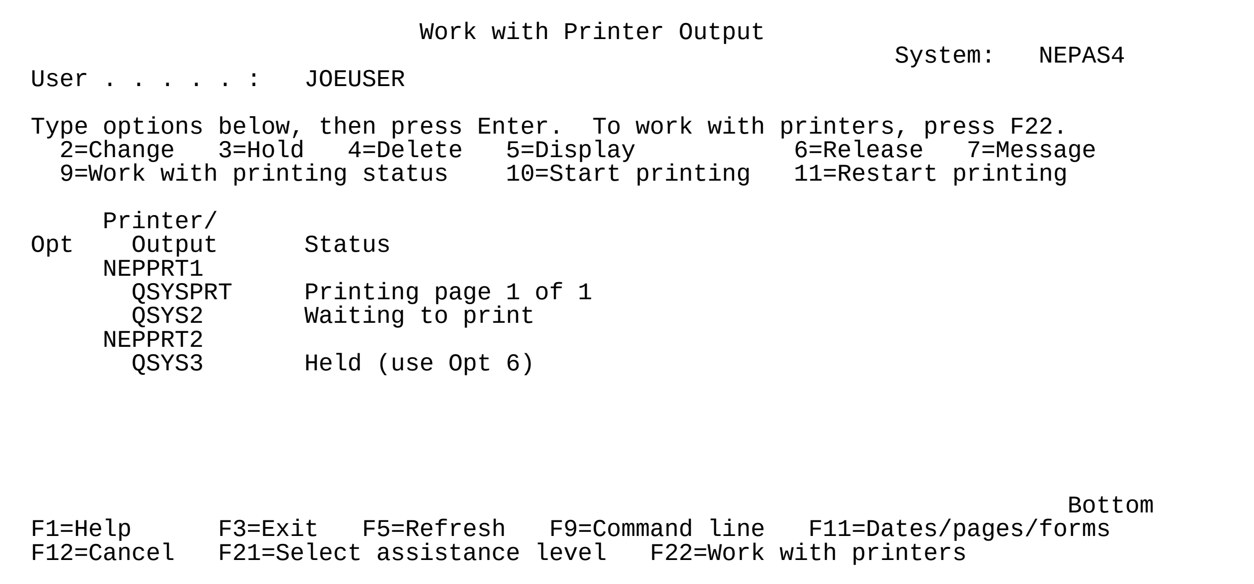
## 16.1.1 Printing a File Member from the Program Development Manager or SEU

When you've listed a file's members with the Program Development Manager, you can print one of these members by entering the number 6 next to that member's name in the `Opt` column. The Program Development Manager will send that member to a print queue, if one has been assigned.

When you finish editing a file member in SEU, the "Exit SEU" screen offers "Print member" among its various options. The default value is "N" for "No," but you can easily Tab your cursor to that field and change it to "Y" if you want to print that member.

## 16.1.2 Checking the Print Queue

As we saw in section 16.1, "Printing Text Files," the Work with Spooled Files (`WRKSPLF`) command lists jobs that are waiting to print. Figure 16.4 shows an example.



```
                         Work with Printer Output
                                                           System:   NEPAS4
User . . . . . :   JOEUSER

Type options below, then press Enter.  To work with printers, press F22.
  2=Change   3=Hold   4=Delete   5=Display          6=Release   7=Message
  9=Work with printing status    10=Start printing   11=Restart printing

     Printer/
Opt    Output      Status
     NEPPRT1
       QSYSPRT     Printing page 1 of 1
       QSYS2       Waiting to print
     NEPPRT2
       QSYS3       Held (use Opt 6)




                                                                    Bottom
F1=Help      F3=Exit   F5=Refresh   F9=Command line   F11=Dates/pages/forms
F12=Cancel   F21=Select assistance level   F22=Work with printers
```

**Figure 16.4 The Work with Printer Output screen.**

Across the top of the screen is the key to the various actions you can take on a waiting print job. We already saw that option 5 shows you what the waiting print job looks like.

Option 2 is a handy one. It displays the "Change Printer Output" screen, which lets you change certain aspects of the print job. One of the most useful fields on the "Change Printer Output" screen is "Printer to Use." This lets you redirect your print job to a different printer.

Options 3 and 6 are also useful. In Figure 16.4, whoever sent job `QSYS3` to the printer has selected 3 to hold the job. It will remain held until that user selects 6 to release the job, allowing it to print. Note how the status message of a held job reminds you that option 6 releases it. Holding a job is useful if you want to select option 2 to change some aspect of how the job is printed. It's also a way to be a nice guy when you have a huge job in front of someone else's smaller job, because holding your job lets the smaller job print first.

## 16.1.3 Canceling Your Print Job

Option 4 of the "Work with Printer Output" screen described in section 16.1.2, "Checking the Print Queue," deletes a print job. If you enter 4 next to one of your jobs and press Enter, it won't be deleted right away; another screen will first prompt you to confirm that you really want to delete that printer job.

## 16.2 Command Files

A stored collection of CL commands that you can execute as a program is called a "CL program." After you use the SEU editor to create a source member of CL commands, it's easy to create a program from that source member and to then use that program from the OS/400 command line.

On most operating systems, the text file that you create as a command file is the same file that you tell the system to run as a program. With OS/400, the file that you create is just the source code for the program that you will run. The source code must be compiled into a program, just like the source code for a program written in C or Pascal. (Don't worry—as we'll see, it's one quick, simple extra step.) This is a great advantage, because a compiled program runs much faster than its text file equivalent.

As we examine the steps in the creation of a CL program and the built-in features of SEU that help us write a CL program, we'll create a sample program that does four things:

1. Run the UpRiteBase database program, telling it to run the report called SUMRPT.
2. Display the report output with the SEU editor, allowing us to make any necessary changes.
3. Print the edited report.
4. Check the print queue to see how long we'll have to wait for the printout.

If this is your first CL program, you'll need to create a file to hold CL programs as members. You can call the file anything you want, but the AS/400 convention for naming a CL program file is `QCLSRC`. Instead of copying another file to create it, you can create it with the Create Source Physical File command:

```
CRTSRCPF FILE(QCSLRC)
```

Start SEU, telling it you want to edit a member in `QCSLRC` called `SUMMARY`.

```
STRSEU QCSLRC SUMMARY
```

Since this member doesn't exist, SEU creates it for you.

All CL programs must begin with the line `PGM` and end with the line `ENDPGM`, so you start the

SUMMARY program with these three lines:

1. PGM
2. The line that tells the URBASE (UpRiteBase) program to run the SUMRPT report.
3. The line that starts the SEU editor up to edit the SUMRPT member of the RPTS file.

It's good programming practice to indent blocks of code, and everything between PGM and ENDPGM is considered a block. Figure 16.5 shows the SEU editor with these three lines added.

```
 Columns . . . :    1  71           Edit                          JOEUSER/QCLSRC
 SEU==>                                                                  CLTEST
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
        *************** Beginning of data ************************************
0001.00 PGM
0002.00              URBASE SUMRPT
0003.00              STRSEU RPTS SUMRPT
        ****************** End of data ***************************************
```

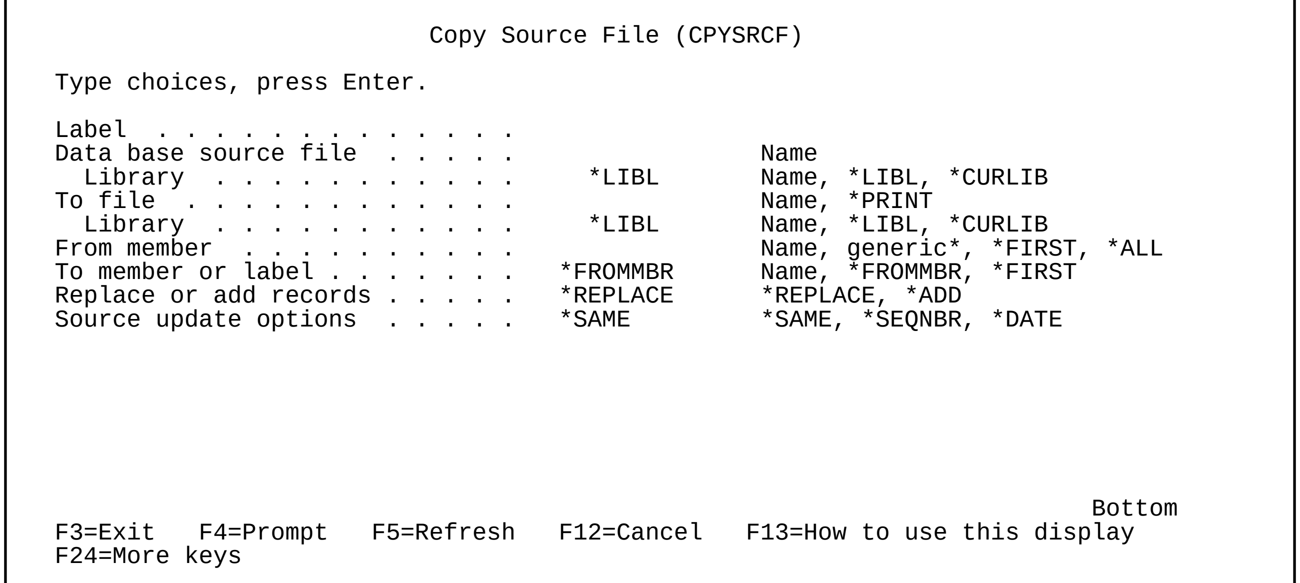
**Figure 16.5 SEU editor with the first three lines of CL program SUMMARY entered.**

Because you are adding a member to a file called QCLSRC, SEU knows that you are entering a CL program, and has two ways to help you:

- SEU points out errors in syntax while you are still composing the CL source program.
- Pressing the F4 key brings up the command prompt display to help you with command syntax the same way that it does when you're entering commands at the command line.

The next line to add to the SUMMARY program shown in Figure 16.5 will print the SUMRPT member with the CPYSRCF command, but what if you can't remember the complete CPYSRCF syntax? If you are entering any command in a CL program and forget the syntax for its parameters, just press F4.

If (after inserting a new blank line with the I line command) you entered CPYSRCF and nothing else on the fourth line of the SUMMARY program and then pressed F4, you would see the screen shown in Figure 16.6—the same screen that you see when you press F4 after entering CPYSRCF at the OS/400 command line.

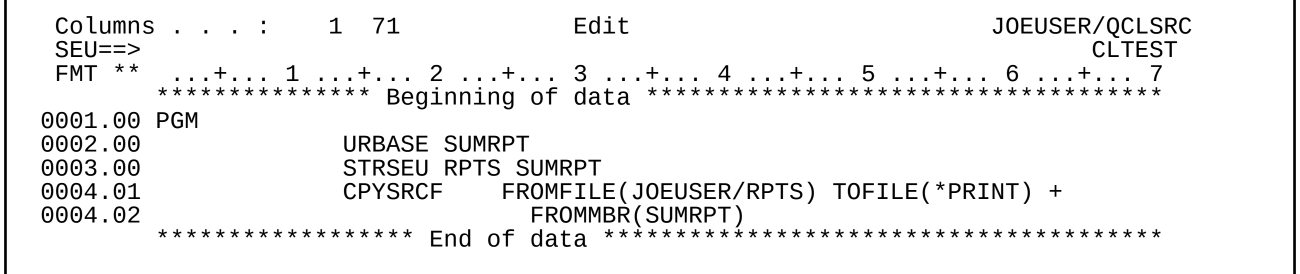


```
                         Copy Source File (CPYSRCF)

 Type choices, press Enter.

 Label  . . . . . . . . . . . . .
 Data base source file  . . . . .                 Name
   Library  . . . . . . . . . . .     *LIBL       Name, *LIBL, *CURLIB
 To file  . . . . . . . . . . . .                 Name, *PRINT
   Library  . . . . . . . . . . .     *LIBL       Name, *LIBL, *CURLIB
 From member  . . . . . . . . . .                 Name, generic*, *FIRST, *ALL
 To member or label . . . . . . .   *FROMMBR      Name, *FROMMBR, *FIRST
 Replace or add records . . . . .   *REPLACE      *REPLACE, *ADD
 Source update options  . . . . .   *SAME         *SAME, *SEQNBR, *DATE








                                                                       Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

**Figure 16.6 Copy Source File command prompt display from within SEU.**

Actually, there is one difference: the optional "Label" field at the top, where you can enter a label to identify that line of the program. In more advanced programming, an instruction of the program could tell the program to jump to that line, and it would use this label to identify the destination of the jump.

After you fill out the screen by entering RPTS as the "Data base source file," *PRINT as "To file," and SUMRPT as the "From member," pressing Enter returns you to the SEU screen. The parameters of the `CPYSRCF` command have been added for you, as shown in Figure 16.7.



```
 Columns . . . :    1  71           Edit                        JOEUSER/QCLSRC
 SEU==>                                                                CLTEST
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
        *************** Beginning of data ************************************
0001.00 PGM
0002.00              URBASE SUMRPT
0003.00              STRSEU RPTS SUMRPT
0004.01              CPYSRCF    FROMFILE(JOEUSER/RPTS) TOFILE(*PRINT) +
0004.02                           FROMMBR(SUMRPT)
        ****************** End of data ***************************************
```

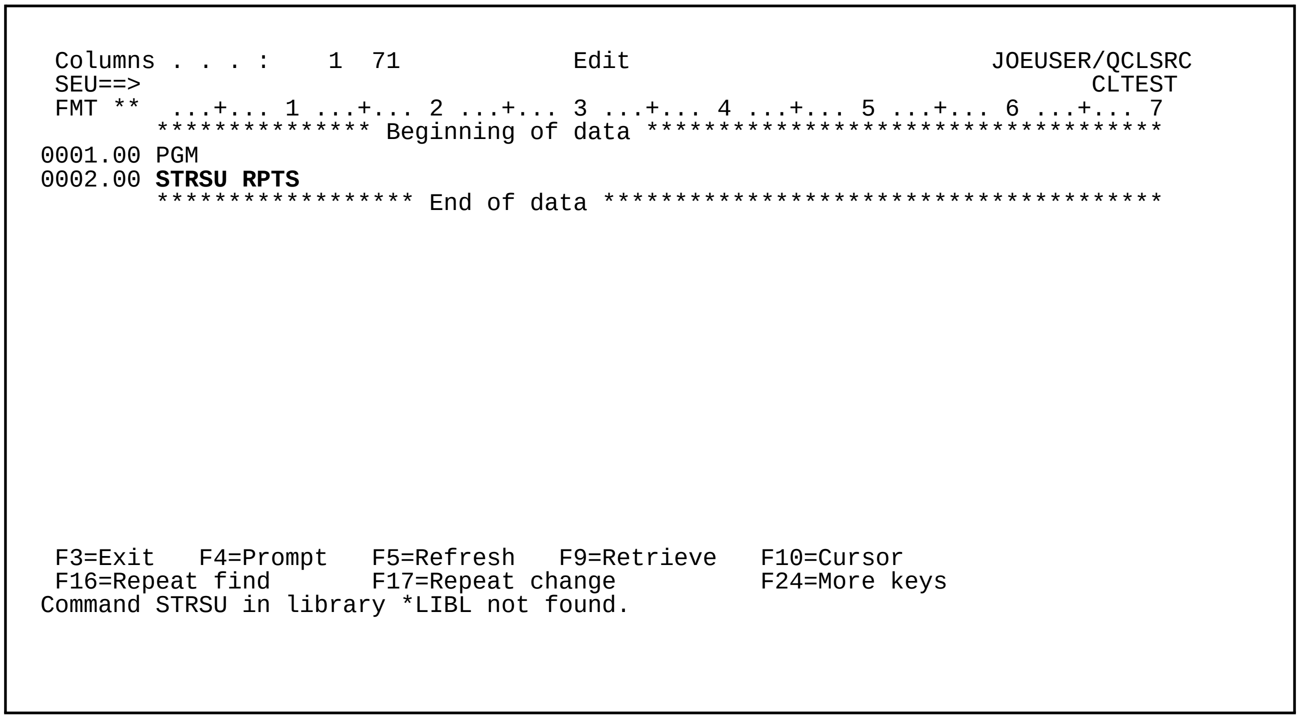
**Figure 16.7 CPYSRCF syntax, automatically filled in by SEU.**

Note how, even if you had entered CPYSRCF on the left side of the screen before pressing F4, it gets indented for you after you fill out the command prompt display.

It also added the parameter names, which makes the line easier to read. However, the indenting plus the parameter names meant that the whole command wouldn't fit on the line. No problem; a plus sign (+) at the end of a line means that the next line is a continuation of that line.

SEU can help you with the syntax of more than just CL programs. It can provide intelligent assistance with the creation of programs written in other languages, like RPG, and even with objects that are not program source code, like the Data Definition Specification (DDS) source code sometimes used as an alternative to IDDU to create databases.

If you make a mistake while entering a CL program, SEU is glad to let you know about it so that you don't have to wait until you attempt to compile the program to find out whether you made a syntax mistake. For example, let's say you forgot the "E" in "STRSEU" and entered "STRSU RPTS SUMRPT" as the third line of the SUMMARY program. After you press Enter, SEU highlights the offending line and displays the same error message at the bottom of the screen that it would have displayed if you had entered STRSU at the OS/400 command line, as shown in Figure 16.8.



```
 Columns . . . :    1  71           Edit                         JOEUSER/QCLSRC
 SEU==>                                                                 CLTEST
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
        *************** Beginning of data ************************************
0001.00 PGM
0002.00 STRSU RPTS
        ****************** End of data ***************************************




 F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F10=Cursor
 F16=Repeat find       F17=Repeat change          F24=More keys
Command STRSU in library *LIBL not found.
```

**Figure 16.8 SEU's reaction to a mistyped "STRSEU" command entered as part of a CL program.**

Once you've properly entered the STRSEU and CPYSRCF commands and then added the final two lines of the CL program, the complete SUMMARY program looks like this:

```
 PGM
         URBASE SUMRPT
         STRSEU RPTS SUMRPT
          CPYSRCF    FROMFILE(JOEUSER/RPTS) TOFILE(*PRINT) +
                FROMMBR(SUMRPT)
         WRKSPLF
```

After saving this file member, you can't run it, because you've only created a member of a source physical file. In other words, you've created the source code for your CL program, not the program itself. Remember, a program is a different kind of object.

To turn your source code into a program, use the Create CL Program command (CRTCLPGM). You only need one parameter for this command: the name that you want to assign to your working CL program. If you omit the name of the file where the source code is stored, CRTCLPGM assumes that it's stored in a file called QCLSRC. If you omit the name of the library where QCLSRC is stored, CRTCLPGM assumes that it will find it in your library list. And, if you omit the name of the member of QCLSRC holding the source code, CRTCLPGM assumes that it has the same name as the program that you are creating. So, if the SUMMARY source member is in the QCLSRC file in a library in your library list, you can compile it into a program by merely typing

```
CRTCLPGM PGM(SUMMARY)
```

or even just this:

```
CRTCLPGM SUMMARY
```

Once it's been created, the Display Library (DSPLIB) command will show SUMMARY as a new object in your library with an object type of *PGM.

To run the program, enter the CALL command followed by the program name. (On some operating systems, every command starts up a corresponding program, but OS/400 makes a distinction between commands and programs. The CALL command is used to run programs.) For the SUMMARY program, this would mean entering this:

```
CALL SUMMARY
```

All the commands entered in the SUMMARY program will be executed one after the other. Each program will also pause for input in the same places that it normally does—for example, when you press F3 to show that you're finished editing the summary report with SEU, the editor displays the "Exit SEU" screen, allowing you to change the parameter values that control whether the file member is printed, whether its lines are renumbered, and so forth.

You can make your CL programs pause for user input. As we saw in section 13.1.2.1, "Command Parameters," entering a question mark (?) before a command and then pressing Enter has the same effect as typing in that command and pressing F4 instead of Enter: it brings up the command prompt display and waits for the user to enter the parameters. This same trick works in CL programs.

For example, if the third line of the `SUMMARY` program had been

```
? STRSEU
```

instead of

```
STRSEU RPTS SUMRPT
```

the `SUMMARY` program would have displayed SEU's command prompt display and waited for the user to enter in the file and member names. After the user was done with SEU, `SUMMARY` would continue with the remaining CL program lines normally.

This adds flexibility to a CL program, because the program's user can fill in different parameters each time while still getting the benefits of an automated series of commands.

### 16.2.1 The Automatic Signon Command File

The Change Profile (`CHGPRF`) command mentioned in section 16.1, "Printing Text Files," has a field on its first display prompt screen called "Initial program to call." Enter the name of a CL program here, and OS/400 will execute that program for you as soon as you sign on, just as if you had entered the `CALL` command to run it as soon as the OS/400 prompt appeared.

Another field on the Change Profile display is called "Initial menu." If you enter a menu name here (remember, each menu has its menu name in the upper-left corner) it will be the first menu to display when you sign on. The `ASSIST` menu (the main menu of the Operational Assistant) is a popular choice for beginners as the initial menu. If you specify both an "Initial program to call" and an "Initial menu" on the Change Profile command prompt display, the specified program will run first and then the system will display the menu when the program is finished.

A popular shortcut for specifying the initial menu is the F23 key. While viewing any OS/400 menu, pressing F23 indicates that you want that menu to be the first menu to appear after signing on. If you look at the Change Profile command prompt display after doing this, you will see that pressing F23 just fills out the "Initial menu" field with the menu's name.

## 16.3 Communicating with Other Users

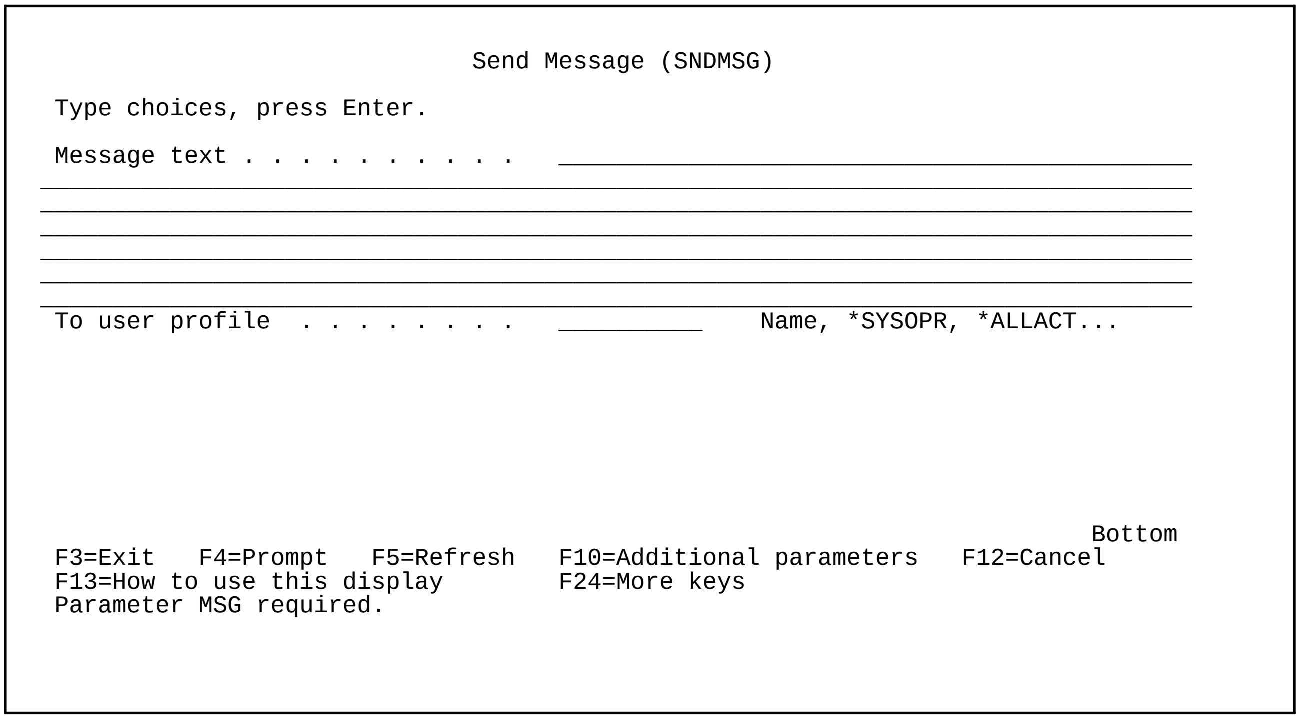
Part of the AS/400's object-oriented approach is its use of messages for communication between the operating system, users, and programs. The operating system communicates to the user via messages—for example, error messages. It communicates to programs via messages—for example, it might tell a program that a given file that it wants to use is not available. The person using this program will be unaware of this communication between the operating system and the program, but the program might be designed to pass along its own version of the same message to the user, because programs can also send messages to users.

And users can send messages to other users just as easily. You can send a message to a mailbox-like storage area called a message queue to wait until the recipient wants to read their waiting messages. In fact, because programs and the operating system can send messages just as users can, you will find messages in your message queue from all three sources.

There are two kinds of messages that you can send:

- *Informational messages*, which simply show up in someone's message queue.
- *Inquiry messages*, which request a reply from the message's recipient.

Send messages with the Send Message (`SNDMSG`) command. After you enter it at the OS/400 command line, pressing F4 or Enter displays the screen shown in Figure 16.9. There are two fields: the long, multi-line one where you enter the message and the one for the name (the signon ID) of the message's recipient.

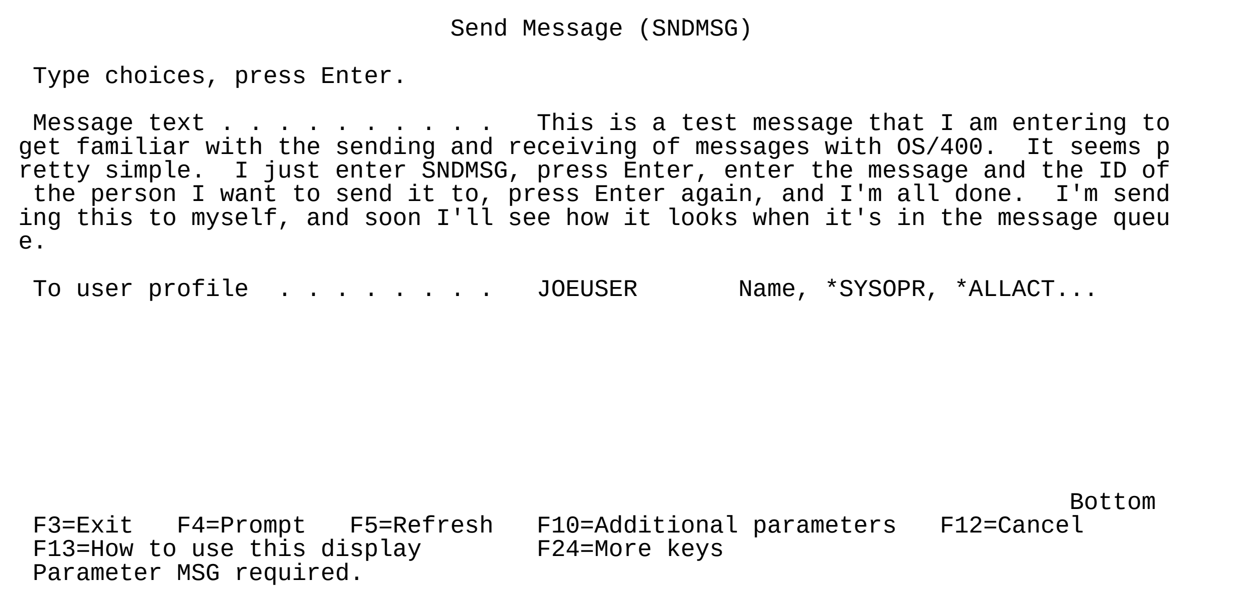


```
                           Send Message (SNDMSG)

 Type choices, press Enter.

 Message text . . . . . . . . . .   ____________________________________________
________________________________________________________________________________
________________________________________________________________________________
________________________________________________________________________________
________________________________________________________________________________
________________________________________________________________________________
________________________________________________________________________________
 To user profile  . . . . . . . .   __________    Name, *SYSOPR, *ALLACT...





                                                                         Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
 F13=How to use this display        F24=More keys
 Parameter MSG required.
```

**Figure 16.9 Entry screen for Send Message command.**

When you enter the message, keep in mind that it will be reformatted. The end of a line on the Send Message command prompt display won't necessarily be the end of the line when the message's recipient reads the message. This means that if a word gets split at the end of the line it won't appear that way to the message's recipient. If you do enter the last character of a word on the end of a line and the first letter of the next word on the beginning of the following line, with no space between the two, they will probably be displayed as one word to the message's recipient.

Figure 16.10 shows a sample message that Joe User has entered. There are three things of interest in this particular message:



```
                                Send Message (SNDMSG)

 Type choices, press Enter.

 Message text . . . . . . . . . .   This is a test message that I am entering to
get familiar with the sending and receiving of messages with OS/400.  It seems p
retty simple.  I just enter SNDMSG, press Enter, enter the message and the ID of
 the person I want to send it to, press Enter again, and I'm all done.  I'm send
ing this to myself, and soon I'll see how it looks when it's in the message queu
e.

 To user profile  . . . . . . . .   JOEUSER       Name, *SYSOPR, *ALLACT...









                                                                          Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
 F13=How to use this display        F24=More keys
 Parameter MSG required.
```

**Figure 16.10 Sample message entered by Joe User to send to himself.**

- Before sending a real message to another user, he wants to get familiar with OS/400's message sending capabilities, so he's entering his own ID as the message recipient.

- The words "pretty," "sending," and "queue" are each split up over two lines. We'll see how they look when Joe "receives" the message. According to what I said above, they shouldn't cause any problem.
- The words "to" at the end of the first line and "get" at the beginning of the second line seem separate enough, but there is no space between them. I warned against this, and we'll see how it looks when the message reaches Joe's message queue.

When you have finished filling out the Send Messages screen, press Enter to show that you are done.

## 16.3.1 Receiving Mail

Use the Display Message (`DSPMSG`) command to display the Work with Messages screen, which shows the messages waiting in your message queue. If Joe User enters `DSPMSG` after he sends the message shown in Figure 16.10, he will see a Work with Messages screen similar to the one shown in Figure 16.11.

```
                              Work with Messages
                                                              System:   NEPAS4
Messages for:   JOEUSER

Type options below, then press Enter.
  4=Remove   5=Display details and reply

Opt   Message
                           Messages needing a reply
      (No messages available)

                         Messages not needing a reply
      This is a test message that I am entering toget familiar with the
        sending and receiving of messages with OS/400.  It seems pretty
        simple.  I just enter SNDMSG, press Enter, enter the message and the
        ID of the person I want to send it to, press Enter again, and I'm all
        done.  I'm sending this to myself, and soon I'll see how it looks when
        it's in the message queue.
        From  . . :   JOEUSER        08/15/93   20:17:16
                                                                        Bottom
F1=Help   F3=Exit   F5=Refresh   F6=Display system operator messages
F16=Remove messages not needing a reply   F17=Top   F24=More keys
```

**Figure 16.11 Message composed in Figure 16.10 as shown by the Display Message command.**

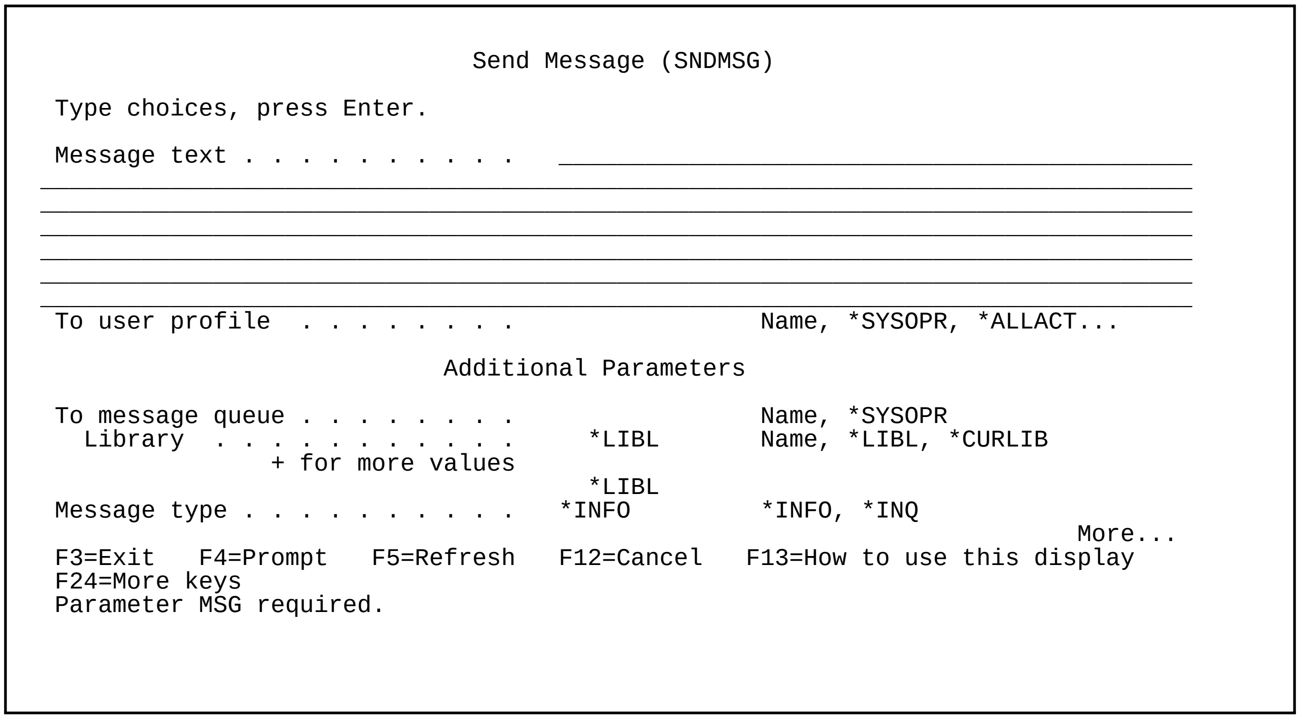
(Another way to display the "Work with Messages" screen is by selecting choice 3 from the Operational Assistant. Type `GO ASSIST` to start up the Operational Assistant.) The upper part of the Work with Messages screen shows any inquiry messages in the queue. In Figure 16.11, the line "(No messages available)" shows that there are none.

In the informational messages ("Messages not needing a reply") part of the screen, note how the words "pretty," "sending," and "queue" are not split up, as they appeared to be when Joe entered the message. The word "to" on the first line is no longer at the end of a line, and since there was no space entered between it and the word "get," they show up in the message as one word: "to-get." Clearly, you need to enter a space after every word, even if a word appears to be separated from the following word by a line break.

Of the actions possible in the `Opt` column, "Display details and reply" only applies to inquiry messages. The only action that Joe can take with this informational message is to enter the number 4 in the `Opt` column, removing the message. If he has several messages and reads them all, F16 provides a shortcut to removing all the messages in the bottom half of the screen at once.

## 16.3.2 Inquiry Messages

Inquiry messages request a response from the recipient. To send one, you enter `SNDMSG`, just like you to do send an informational message. On the "Send Message" screen shown in Figure 16.10, note how the F10 key will display "Additional Parameters." Pressing it when viewing a blank "Send Message" screen adds several fields to the screen, as shown in like Figure 16.12.



```
                              Send Message (SNDMSG)

 Type choices, press Enter.

 Message text . . . . . . . . . .   ____________________________________________
________________________________________________________________________________
________________________________________________________________________________
________________________________________________________________________________
________________________________________________________________________________
________________________________________________________________________________
________________________________________________________________________________
 To user profile  . . . . . . . .                 Name, *SYSOPR, *ALLACT...

                           Additional Parameters

 To message queue . . . . . . . .                 Name, *SYSOPR
   Library  . . . . . . . . . . .     *LIBL       Name, *LIBL, *CURLIB
              + for more values
                                      *LIBL
 Message type . . . . . . . . . .   *INFO         *INFO, *INQ
                                                                      More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
 Parameter MSG required.
```

**Figure 16.12 "Send Message" screen with additional parameters displayed.**

The "More..." message at the bottom of the screen shows that there are more parameters than will fit, but the one we want is there on the screen: "Message type." The default value is *INFO, for "informational message." For an inquiry message, enter *INQ. If you set "Message type" to *INQ, the message that you enter will appear on the recipient's "Work with Messages" screen under "Messages needing a reply." (If you know that you want to send an inquiry before you enter `SNDMSG` at the OS/400 command line, you can add the parameter `MSGTYPE(*INQ)` to the `SNDMSG` command. Except for very short messages, AS/400 users generally don't enter all `SNDMSG` parameters at the OS/400 command line because it's easier to edit the actual message on the `SNDMSG` command prompt display than to include it on the OS/400 command line.)

To see how this works, Joe sets "Message type" to *INQ, enters his own ID in the "To user profile" field, enters a message of "How will this look?", and presses Enter. When the OS/400 command line reappears, he enters `DSPMSG` and sees the screen shown in Figure 16.13.

```
                              Work with Messages
                                                            System:   NEPAS4
Messages for:   JOEUSER

Type options below, then press Enter.
  4=Remove   5=Display details and reply

Opt   Message
                           Messages needing a reply
      How does this look?
        From  . . :   JOEUSER        08/17/93   10:04:57

                         Messages not needing a reply
      (No messages available)





                                                                        Bottom
F1=Help   F3=Exit   F5=Refresh   F6=Display system operator messages
F16=Remove messages not needing a reply   F17=Top   F24=More keys
```

**Figure 16.13 Work with Messages display of a message needing a reply.**

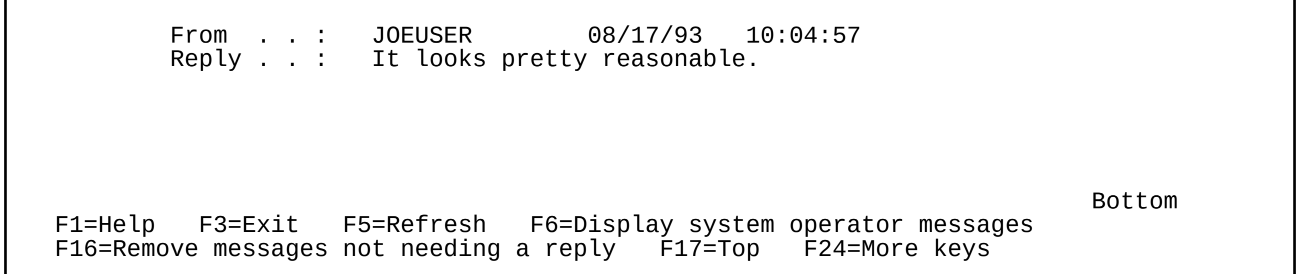
He replies to the message by entering a 5 in the `Opt` column next to the message and pressing Enter. The Display Message program displays the "Additional Message Information" screen for an inquiry message, which tells him to "Type reply below" at the bottom of the screen. Joe enters the response shown in Figure 16.14 and presses Enter.

```
                         Additional Message Information

 From . . . . . . . . . . :   JOEUSER
 Date sent  . . . . . . . :   08/17/93      Time sent  . . . . . . . :   10:04:57

 Message . . . . :   How does this look?










                                                                             Bottom
 Type reply below, then press Enter.
 Reply  . . . .   It looks pretty reasonable.
```

**Figure 16.14 Responding to an inquiry message.**

After he responds to the inquiry, he sees the "Work with Messages" screen again, and it has changed: because he has responded to the "How does this look?" message, it no longer needs a reply. It is now displayed, with its answer, under "Messages not needing a reply," as shown in Figure 16.15.

```
                               Work with Messages
                                                           System:   NEPAS4
 Messages for:   JOEUSER

 Type options below, then press Enter.
   4=Remove   5=Display details and reply

 Opt   Message
                             Messages needing a reply
       (No messages available)

                           Messages not needing a reply
       How does this look?
```



```
         From  . . :   JOEUSER        08/17/93   10:04:57
         Reply . . :   It looks pretty reasonable.





                                                                       Bottom
 F1=Help   F3=Exit   F5=Refresh   F6=Display system operator messages
 F16=Remove messages not needing a reply   F17=Top   F24=More keys
```

**Figure 16.15 Work with Messages screen after response to inquiry message.**

Do the inquiry message ("How does this look") and the response now appear under "Messages not needing a reply" because Joe was the originator of the inquiry or because he was the recipient? The answer is both. After an inquiry recipient responds to the inquiry, the inquiry and its response appear under "Messages not needing a reply" for both the sender and the recipient to give them a record of their exchange. The exchange remains there until they delete it by entering the number 4 in the `Opt` column next to it or by pressing F16.

## 16.3.3 Sending an Existing File

The Send Network File (`SNDNETF`) command can send file members to other users on your AS/400 or, as its name implies, to other users on systems attached to your AS/400 over a network—even if the system they are using is not an AS/400.

To send the `SUMRPT` member of the `RPTS` file to Mary Jones' MJONES user ID on the JUPITER system, Joe User enters the following:

```
SNDNETF FILE(RPTS) TOUSRID((MJONES JUPITER)) MBR(SUMRPT)
```

If he omitted the parameter names and entered the positional parameter version, he would enter this:

```
SNDNETF RPTS ((MJONES JUPITER)) SUMRPT
```

There are several interesting things to note about this command:

- If RPTS wasn't in a library in Joe's library list, he would have to specify which library it was in with the `(libname/filename)` syntax.

- The destination user ID and system name are enclosed together in two pairs of parentheses, even in the positional parameter version of the command.

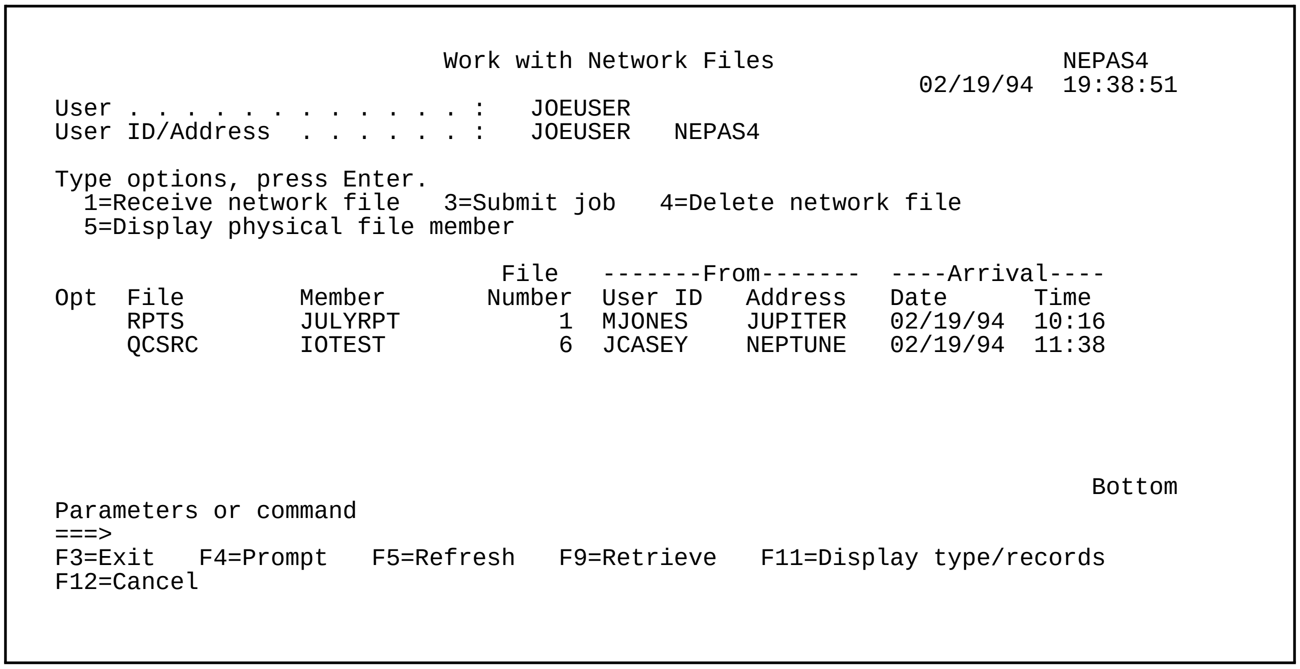
- The system name `JUPITER` might represent an AS/400, but it might not. The system name of your particular AS/400 will be displayed on the signon screen; in Figure 13.1, it's `NEPAS4`. This means that someone sending a file to Joe from their AS/400 ID would send it to `((JOEUSER NEPAS4))`. (From a UNIX system connected over the same network, they would send it to `JOEUSER@NEPAS4`, and from a VM ID it would be addressed to `JOEUSER AT NEPAS4`.)

- Unlike similar commands on other computers, you must specify the destination system name even if you send it to an ID on the same system that you are using. (Other operating systems usually assume "same system" as a default if you omit the system name when identifying a recipient.)

- Don't take it for granted that you have permission to use `SNDNETF`. If you try it and get the message "User not enrolled in system distribution directory," speak to your system administrator about being enrolled to use the command.

### 16.3.3.1 Receiving a File

The Work with Network Files command (`WRKNETF`) displays the screen shown in Figure 16.16. This lets you look at and receive the files that have been sent to you with `SNDNETF` or with the equivalent command on other computers connected to your system over a network. (Like `SNDNETF`, you must be enrolled by the system administrator to use `WRKNETF`.)



```
                           Work with Network Files                    NEPAS4
                                                             02/19/94  19:38:51
User . . . . . . . . . . . . :   JOEUSER
User ID/Address  . . . . . . :   JOEUSER   NEPAS4

Type options, press Enter.
  1=Receive network file   3=Submit job   4=Delete network file
  5=Display physical file member

                               File   -------From-------  ----Arrival----
Opt  File        Member       Number  User ID   Address   Date      Time
     RPTS        JULYRPT           1  MJONES    JUPITER   02/19/94  10:16
     QCSRC       IOTEST            6  JCASEY    NEPTUNE   02/19/94  11:38





                                                                       Bottom
Parameters or command
===>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F11=Display type/records
F12=Cancel
```

**Figure 16.16 The Work with Network Files screen displayed by the WRKNETF command.**

The F11 key changes the display so that the type and number of records of the waiting files are shown instead of the `User ID` and `Address` columns. Figure 16.17 shows an example.

```
                           Work with Network Files                    NEPAS4
                                                            02/19/94  19:38:51
User . . . . . . . . . . . . . :   JOEUSER
User ID/Address  . . . . . . . :   JOEUSER   NEPAS4

Type options, press Enter.
  1=Receive network file   3=Submit job   4=Delete network file
  5=Display physical file member

                                                    Record  -----Send------
Opt  File        Member      Type        Records  Length  Date      Time
     RPTS        JULYRPT     *DTA             62      29  02/19/94  18:15
     QCSRC       IOTEST      *SRC            182      92  02/19/94  19:38





                                                                      Bottom
Parameters or command
===>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F11=Display user ID/address
F12=Cancel
```

**Figure 16.17 The Work with Network Files screen with object type, number of records, and record length of the waiting files displayed.**

In Figure 16.17, we see that the `JULYRPT` member of the `RPTS` file is a simple data file with 62 lines of text. The other file is source code for a program (judging from the file name, it's written in C) and it's about three times longer than `JULYRPT`.

The command key at the top shows that entering a 1 in the `Opt` column will store the waiting member in a file. When you do this, the system assumes that you already have a file with the name shown in the `File` column for that member, and it stores the member there. If you don't have a file by that name, press F4 instead of Enter after entering the 1 in the `Opt` column. This displays the Receive Network File command prompt display, where you can enter any file name you like.

## 16.4 A Sample OS/400 Session

One morning you sign on to your OS/400 ID and check your messages with the `DSPMSG` command, and you see the screen shown in Figure 16.18.

```
                              Work with Messages
                                                            System:   NEPAS4
Messages for:   JOEUSER

Type options below, then press Enter.
  4=Remove   5=Display details and reply

Opt   Message
                           Messages needing a reply
      You said that you know how to write CL programs, right?  I need one
        for the guys in the warehouse.  We're going to send them a file
        with a list of orders over the network each day, and I don't want
        to have to teach them about WRKNETF and printing.  The member
        will be called ORDERS and go into a file called INVEN (they
        already have this file, so don't worry about creating it).  When
        they type CALL GETORDERS, I want this program to get ORDERS out
        of the queue of network files and send it to their printer,
        which is called WRHOUSE.  Can you handle this?
        From  . . :   MARYJONES     04/27/94   10:03:22

                         Messages not needing a reply
      (No messages available)
                                                                        Bottom
F1=Help   F3=Exit   F5=Refresh   F6=Display system operator messages
F16=Remove messages not needing a reply   F17=Top   F24=More keys
```

**Figure 16.18 Message from Mary about CL program to write.**

You type in a 4 next to the first line of Mary's message, press Enter, and enter "No problem, Mary, you'll have it by this afternoon" as your reply.

But there is a slight problem: you know how to receive a network file with `WRKNETF`, but how would you automate this? You use `WRKNETF` by making choices on a list display; you can't tell it what to do by entering `WRKNETF` on the OS/400 command line with a couple of parameters after it. It would be a good idea to review the process of receiving a file, so you send yourself a short little file—the source code to the first C program that you wrote on the AS/400—over the network with the following command:

```
SNDNETF FILE(QCSRC) TOUSRID((JOEUSER NEPAS4)) MBR(CTEST1)
```

You then enter `WRKNETF` to bring up the Work with Network Files screen and enter a 1 next to the `CTEST1` member of the `QCSRC` file. Instead of pressing Enter, you press F4 to see how much control you can have over the receiving of a file. This displays the Receive Network File screen, as shown in Figure 16.19.

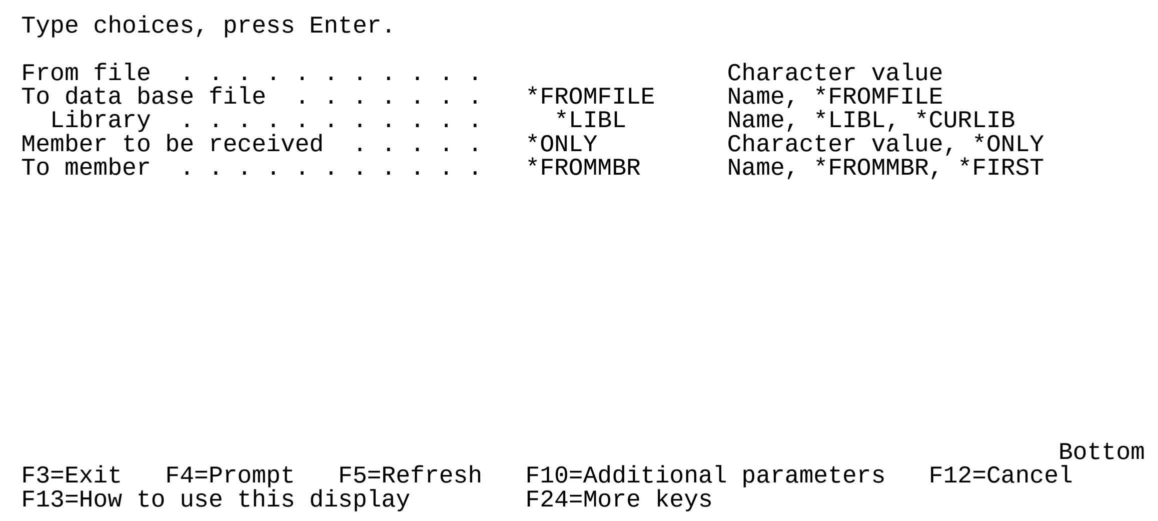
```
                         Receive Network File (RCVNETF)

 Type choices, press Enter.

 From file  . . . . . . . . . . . > QCSRC         Character value
 To data base file  . . . . . . . > *FROMFILE     Name, *FROMFILE
   Library  . . . . . . . . . . .     *LIBL       Name, *LIBL, *CURLIB
 Member to be received  . . . . . > CTEST1        Character value, *ONLY
 To member  . . . . . . . . . . .   *FROMMBR      Name, *FROMMBR, *FIRST













                                                                      Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
 F13=How to use this display        F24=More keys
```

**Figure 16.19 Receive Network File command prompt display, as displayed by WRKNETF program.**

The `RCVNETF` after the screen's title makes you realize something: when you entered the 1 at the Work with Network Files screen, it just called a program called `RCVNETF`. You could have your CL program call `RCVNETF` directly, and skip over the Work with Network Files screen.

First, you should test this idea. You press F3 a couple of times to go back to the command line, enter `RCVNETF` by itself, and press F4. When you see the screen shown in Figure 16.20, you recognize it as the same screen that `WRKNETF` displayed when you entered a 1 in the `Opt` column. So `WRKNETF` really was just calling `RCVNETF`!

```
                         Receive Network File (RCVNETF)
```



```
Type choices, press Enter.

From file  . . . . . . . . . . .                      Character value
To data base file  . . . . . . .   *FROMFILE     Name, *FROMFILE
  Library  . . . . . . . . . . .     *LIBL       Name, *LIBL, *CURLIB
Member to be received  . . . . .   *ONLY         Character value, *ONLY
To member  . . . . . . . . . . .   *FROMMBR      Name, *FROMMBR, *FIRST










                                                                          Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

**Figure 16.20 Receive Network File command prompt display, as displayed by entering RCVNETF command.**

To receive the file that you sent yourself as a test, you enter "QCSRC" in the "From file field," "CTEST1" in the "Member to be received" field, and press Enter. The system returns to the main menu, and displays the message

```
File QCSRC member CTEST1 number 14 received.
```

at the bottom. It worked! You press F9 to retrieve the command that you would have typed at the command line, and you see this:

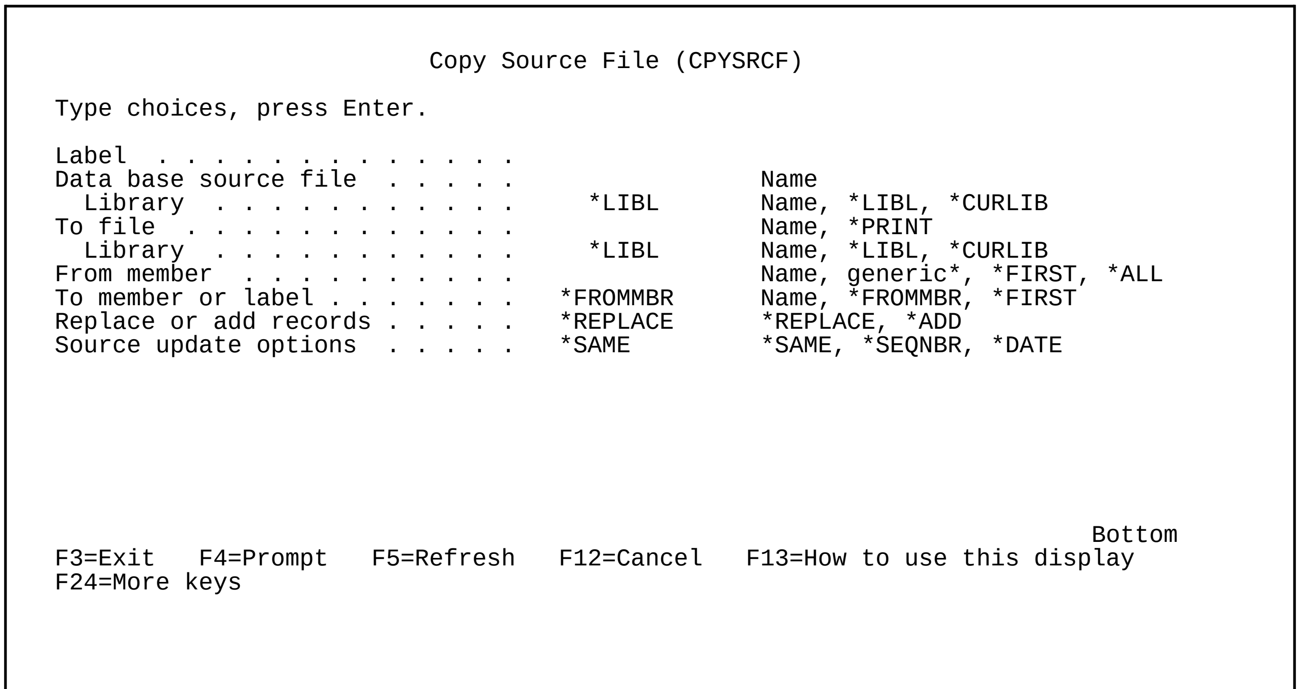
```
RCVNETF FROMFILE(QCSRC) FROMMBR(CTEST1)
```

So that's the command that you'll need for your `GETORDERS` CL program, only with a file name of `INVEN` and a member name of `ORDERS`. So, you start up SEU and tell it to create the `GETORDERS` member of your `QCSLSRC` file with the following command:

```
STRSEU QCLSRC GETORDERS
```

You enter `PGM` as the program's first line, because that's the first line of the source code in all CL programs, and the `RCVNETF` command as the second line, with `INVEN` and `ORDERS` as its parameters.

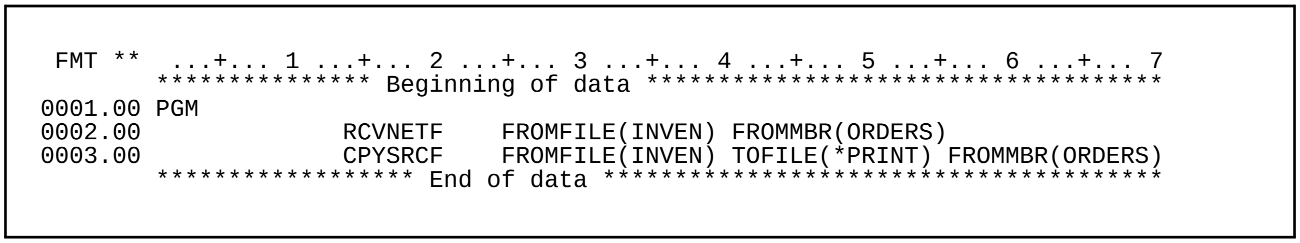
The `GETORDERS` program's next task is to print the newly received member. You remember that this is done with some trick using the `CPYSRCF` command, but you don't remember the exact syntax. You enter `CPYSRCF` as the third line of the CL program's source code and press F4 to find out more about its potential parameters. SEU displays the screen shown in Figure 16.21.



```
                          Copy Source File (CPYSRCF)

 Type choices, press Enter.

 Label  . . . . . . . . . . . . .
 Data base source file  . . . . .                 Name
   Library  . . . . . . . . . . .     *LIBL       Name, *LIBL, *CURLIB
 To file  . . . . . . . . . . . .                 Name, *PRINT
   Library  . . . . . . . . . . .     *LIBL       Name, *LIBL, *CURLIB
 From member  . . . . . . . . . .                 Name, generic*, *FIRST, *ALL
 To member or label . . . . . . .   *FROMMBR      Name, *FROMMBR, *FIRST
 Replace or add records . . . . .   *REPLACE      *REPLACE, *ADD
 Source update options  . . . . .   *SAME         *SAME, *SEQNBR, *DATE








                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

**Figure 16.21 Copy Source File command prompt display from within SEU.**

Once you see the displayed screen, you know that you enter `INVEN` as the "Data base source file" and `ORDERS` as the "From member." Looking to the right of "To file," you see that *PRINT is a possible option, instead of "Name." So that was the trick. Entering a name would have copied the member to another member with the specified name, but entering *PRINT sends it to the printer. You enter "*PRINT," press Enter, and the system returns you to the SEU screen where it has filled out the `CPYSRCF` line for you, as shown in Figure 16.22.



```
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
        *************** Beginning of data ************************************
0001.00 PGM
0002.00               RCVNETF    FROMFILE(INVEN) FROMMBR(ORDERS)
0003.00               CPYSRCF    FROMFILE(INVEN) TOFILE(*PRINT) FROMMBR(ORDERS)
        ****************** End of data ***************************************
```

**Figure 16.22 First three lines of the GETORDERS CL program's source code.**

Most people want to see the print queue after they send something to the printer, to determine whether they should walk to the printer right away to get their print job or if they should wait because of some other print jobs in front of theirs. You add `WRKSPLF` as the fourth line of `GETORDERS` so that the print queue gets automatically displayed, and then `ENDPGM` to finish the source code. At this point, your program looks like Figure 16.23.

```
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
        *************** Beginning of data ************************************
0001.00 PGM
0002.00              RCVNETF    FROMFILE(INVEN)
0003.00              CPYSRCF    FROMFILE(INVEN) TOFILE(*PRINT) FROMMBR(ORDERS)
0004.00              WRKSPLF
0005.00 ENDPGM
        ****************** End of data ***************************************
```

**Figure 16.23 Finished source code for GETORDERS.**

You quit out of SEU, saving the file, and enter the command to compile this source code into a working program:

```
CRTCLPGM GETORDERS
```

When you see the message "Program GETORDERS created in library JOEUSER," you know that it successfully compiled. But you still have to test it; you create a file called `INVEN` with a dummy member named `ORDERS`. You send a copy of `ORDERS` to your message queue with the command

```
SNDNETF FILE(INVEN) TOUSRID((JOEUSER NEPAS4)) MBR(ORDERS)
```

and then delete `ORDERS` from the `INVEN` file with the `RMVM` command. This way, if your program works correctly and pulls `ORDERS` out of the message queue and puts it in `INVEN`, you'll know that it was put there by `GETORDERS` and not left over from the original one that you created.

Time for the big test. You enter

```
CALL GETORDERS
```

and press Enter. Shortly after that, you're looking at the Work with Network Files screen, and there's the print job sent by your `GETORDERS` program. You press F3 to return to the command line and list the members in `INVEN` by entering

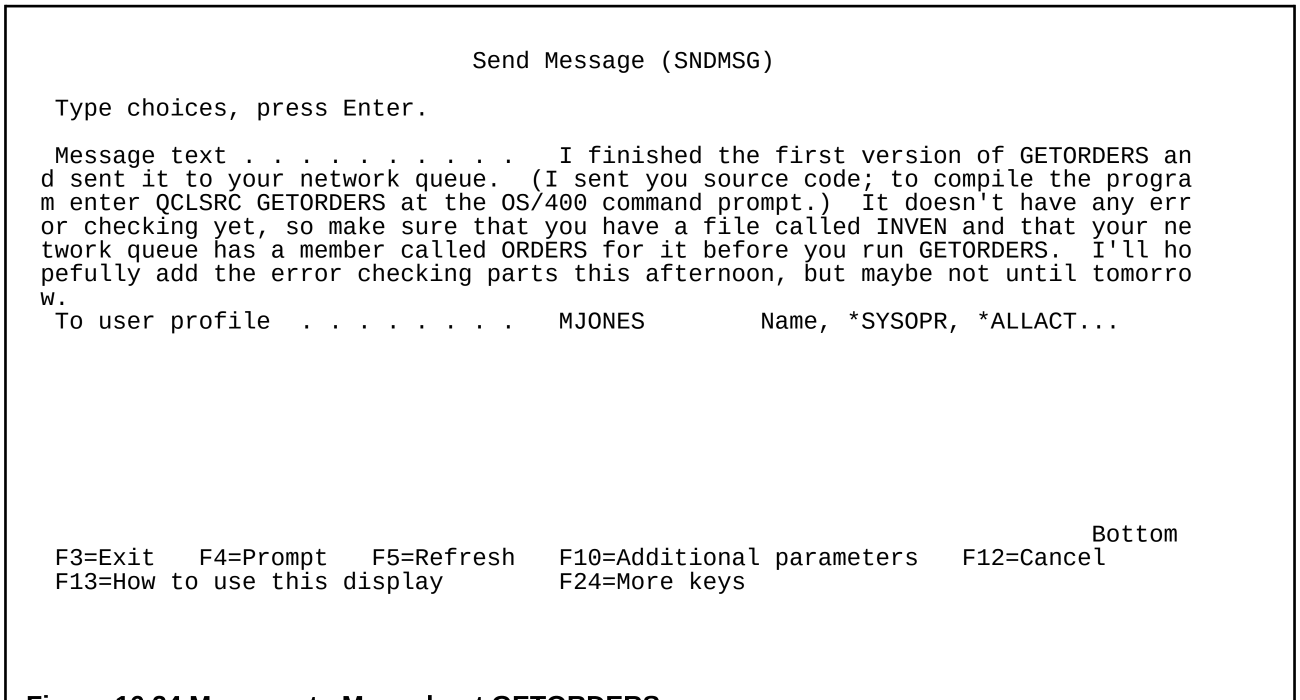
```
DSPFD INVEN TYPE(*MBRLIST)
```

and there is `ORDERS`, even though you deleted it a minute ago. Now you know that `GETORDERS` put it there like it was supposed to.

It looks like the `GETORDERS` program works, so you send it to Mary with the command

```
SNDNETF FILE(QCLSRC) TOUSRID((MJONES NEPAS4)) MBR(GETORDERS)
```

and then use `SNDMSG` to send the message shown in Figure 16.24 to her as well.



```
                              Send Message (SNDMSG)

  Type choices, press Enter.

  Message text . . . . . . . . . .   I finished the first version of GETORDERS an
 d sent it to your network queue.  (I sent you source code; to compile the progra
 m enter QCLSRC GETORDERS at the OS/400 command prompt.)  It doesn't have any err
 or checking yet, so make sure that you have a file called INVEN and that your ne
 twork queue has a member called ORDERS for it before you run GETORDERS.  I'll ho
 pefully add the error checking parts this afternoon, but maybe not until tomorro
 w.
  To user profile  . . . . . . . .   MJONES        Name, *SYSOPR, *ALLACT...








                                                                              Bottom
  F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
  F13=How to use this display        F24=More keys
```

**Figure 16.24 Message to Mary about GETORDERS.**

Now all that's left is the error checking. You enter `STRSCHIDX` to start up the search index and then enter "CL program" as the search string, confident that you will find what you need in the OS/400's extensive help system.