

---

## Communicating with the System

### Overview

The IBM i operating system (OS) is one of the most sophisticated on the market today. To a new user, the apparent complexity may seem like a huge jigsaw puzzle waiting to be solved. What we hope to provide with this first chapter is simply a place to start collecting critical pieces of the puzzle.

This chapter provides initial exposure to several topics necessary for a basic understanding of how to communicate with the IBM i OS using the different types of screens and menus. The chapter also introduces concepts related to system organization and a few basic **work management** ideas (e.g., how jobs enter, run in, and leave the system).

As in any other technical field, you will need to master a number of terms and concepts to be successful using the OS. Although the IBM i OS will provide comprehensive help as you learn, you are strongly encouraged to ask questions of your instructor or mentor.

### Objectives

Students will be able to

- Sign on to the IBM i OS system, properly entering a user profile and password
- Explain the difference between system and subsystem
- Describe the two most common types of jobs
- List several attributes of a user profile
- List several features of the IBM i OS
- Explain the function of IBM i Control Language (CL) and enter a CL command on a command line using proper syntax

*Continued*

- Explain the purpose of system values and how to check them
- Describe four types of displays, their use, and their components
- Explain the relationship between CL commands and IBM i OS menu paths

## The System

The **IBM i OS** is a complex and rich set of programs that not only controls traditional functions such as data access, storage, and task management but also incorporates features that normally would require separate software components on other systems—features such as communications support, database management, security, and interactive support.

A new system running IBM i OS comes with a standard basic configuration, which includes predefined system settings for work management functions that allow interactive and batch jobs to run without need for a customized installation. Later, subsystems can be created and system resources allocated to optimize workload distribution and throughput according to the special needs of a particular business.

IBM i is a **multiuser, multitasking** OS (a system on which two or more people can perform two or more tasks concurrently) optimized for the efficient execution of business applications. The basic unit of work is a **job**. The term *job* refers generally to a unit of work that includes all programs, files, and instructions necessary to perform that work. Examples of a job would be an interactive user session for updating a customer master file, or a program compilation running unobtrusively in the background.

The older IBM AS/400 systems used terminals and hardwired (twinaxial, or “twinax”) connections to access the system. Today, most users connect to the system using 5250 emulators or Web interfaces. In this book, we present a number of ways to connect to the system, including 5250 emulators, Remote System Explorer (RSE)/Rational Developer for Power (RDP)-based, and IBM i Access for Web and Access for Windows. The IBM i OS has evolved to become one of the most advanced OSs available today.

## IBM i Objects

The IBM i OS is an object-based OS. An **object** is anything on the system that has a name and takes up space in storage. IBM supplies a large number of objects through the IBM i OS and licensed program products. These objects typically have names beginning with *Q*. Programmers and operators create other objects and name them according to their organizations’ naming conventions.

The system can locate an object by its name; once located, the object can further identify itself to the system by functional attributes that are a part of the object. Objects are grouped into types. An **object type** determines how the object is used on the system (i.e., the actions that one can take when using the object). Common object types include programs, files, and commands. Object types also include user-profile objects, which contain information about users, and subsystem-description objects, which contain the characteristics of subsystems. Object type is always assigned by the system and is determined by the command used to create the object. We continue to explore this concept of objects throughout this book.

## Subsystems

All jobs in the IBM i OS are run in **subsystems**. A subsystem is a predefined operating environment the system uses to coordinate workflow and system resources. The system uses a **subsystem description** to define the required resources to process work. The components of a subsystem description determine how the system uses resources to process jobs within the subsystem. When the base IBM i OS is installed, several different subsystems are already defined and active, and each one has a separate subsystem description.

Attributes of subsystems are defined in the subsystem description; these include the subsystem name (e.g., **QCTL** for the controlling subsystem, **QINTER** for the interactive subsystem, **QBATCH** for the batch subsystem), how many jobs can run in the subsystem at one time, which storage pools the subsystem will use (IBM i main storage is divided into a number of different storage pools), and which job queues the subsystem will use.

Different subsystems are necessary because there are many different types of jobs with different characteristics and often-conflicting needs. If we treated all of these various jobs equally, the overall performance of the system would suffer. As system administrators, we can change the existing subsystem descriptions and create new subsystems when necessary. In this way, we can tailor subsystems to efficiently handle the needs of different jobs—for example, to ensure that long-running jobs that require no user interaction but significant CPU time do not interfere with high-priority interactive jobs that need fast response time. Within subsystems, we can prioritize individual jobs to begin execution sooner or later; and, after they begin execution, we can give them a higher or lower runtime priority.

## Types of Jobs

IBM i OS jobs can originate from several sources and are classified by how they originate on the system. In this course, and in general, you will deal mostly with two types of jobs: *interactive* and *batch*.

An **interactive job** begins when a user signs on to the system, and it terminates when the user signs off the system or the job is ended. Interactive jobs run in conversational mode, which means there is a dialogue of sorts between the user and the program, utility, or OS function. Because of this conversational, back-and-forth nature of interactive jobs, any CPU- or I/O-intensive request a user makes could lock up the workstation keyboard until the request is completed. Therefore, it is often advisable to direct such requests to a subsystem designed to handle them—that is, to submit them as batch jobs.

**Batch jobs** can execute without user intervention; they do not require data or controlling values to be input through the workstation once they have started. Batch jobs are sent to a job queue until they can begin execution. A **job queue** is a staging area, managed by the subsystem in which the job will run, where batch jobs wait in line for their turn at processing. Each batch subsystem can execute only a limited number of batch jobs concurrently. If no other higher-priority jobs are waiting, a batch job can start right away; otherwise, it must wait its turn.

Typically, you would submit as a batch job a report program that reads many records from a database file and performs standard calculations written into the logic of the program. Once the program is submitted, it requires no input from the operator or requester of the report. If the program were run interactively, the direct-access storage device (**DASD**, or **hard disk**), access time required for the large file would be substantial and could cause the program to run for an extended period, tying up the workstation until program execution was completed.

In addition, to minimize disruption of workflow, programmers can pack up and send off as batch jobs certain tasks they encounter during a normal interactive session. Consider a programmer in the midst of an interactive session who needs to compile a large program for testing. If she runs the compile as part of an interactive job, she will be locked out of the system during the several minutes it may take to complete the compilation. If other tasks on the computer need attention, she could submit the compilation as a batch job from within the interactive job. Then, while the compilation runs in a batch subsystem, she could go on to other tasks.

## **Control Language (CL)**

For all its complexity, one of the more remarkable things about the IBM i OS is that it provides a single, consistent user interface to its various functions through Control Language (**CL**) commands.

IBM i OS CL is flexible and powerful and allows direct access to IBM i OS functions. More than 2,000 individual CL commands are available. Each individual command is an OS object. Most CL commands consist of a command name and one or more **command parameters**. A command parameter is a value, specified along with a command, that controls and limits the operation of the command and names the files, programs, or other objects the command will work on. Individual commands can be entered on a **command line** (a line beginning with the symbol `====>` that appears near the bottom of certain types of display screens), or they can be grouped together into a **CL program** to perform a specific task. Such a program can then be compiled and run from a command line or from within a high-level language (**HLL**) program (e.g., RPG, COBOL, C, or C++).

If you are a new user, you don't need to memorize a large number of CL commands because most IBM i OS functions are available to interactive jobs through the **menu interface** of IBM i. As you choose a series of menu selections, the system determines the CL commands to be run to satisfy your request.

## System Values

**System values** are control and configuration attributes that let you customize certain OS functions. System values define critical aspects of the environment and the general rules that all jobs on the system must follow. System values are not objects, but they describe characteristics of the system that can be displayed, or in many cases changed, with CL commands. Many system values come preset with the OS; others need to be set when the system is installed. The categories of system values include the following:

- *Allocation values* (Type=\*ALC)—These values let you control the number of active jobs and how much main storage will be used for different functions necessary to run jobs.
- *Date and time values* (Type=\*DATTIM)—These values let you set and change the date and time the system keeps and makes available to application programs and utilities.
- *Editing values* (Type=\*EDT)—These values control how dates, decimal values, and numbers involving currency symbols are formatted.
- *Library-list values* (Type=\*LIBL)—These values define the system library list and initial user library list. A **library list** is an ordered group of libraries a job uses to search for objects it needs for processing. A **library** is a directory of related objects.
- *Message and logging values* (Type=\*MSG)—These values control how the system handles and records certain types of messages.

- *Security values* (Type=\*SEC)—These values control certain aspects of security, such as the maximum number of invalid sign-on attempts allowed.
- *Storage system values* (Type=\*STG)—These values define the minimum size and activity level (number of active jobs) of the base storage pool.
- *System control values* (Type=\*SYSCTL)—These values let you define or obtain certain controlling values of the OS, such as operator console name, user assistance level, and date and time to automatically **IPL** the system. (An IPL, or initial program load, is the IBM i version of a “boot” process, which loads the OS when the power is turned on.)

Most users can display current system values, and, when necessary, authorized users can change these values. This capability to customize system values for a particular job environment can lead to a more efficiently running system. In the lab for this chapter, you learn how to examine system values.

## Licensed Programs

Besides the IBM i OS, several software components available from IBM extend the functions and capabilities of the system and let installations tailor the system to the type of work and the communications environment they need. These software components are **licensed program products**. A few licensed program products are shipped with all IBM i systems at no additional cost. But for the most part, licensed program products are chargeable; the actual cost usually is determined by the size of the system and the number of users who have access to the product. Most of the licensed program products fall into one of several categories:

- *Application development tools*—This group of programmer utilities greatly expedites the application development process. Recently, IBM has brought its software tools under the Rational brand. Rational Developer for Power (**RDP**) is a set of GUI-based tools designed to replace traditional 5250 tools such as Source Entry Utility (**SEU**), Screen Design Aid (**SDA**), an interactive file-update utility called Data File Utility (**DFU**), and a programmer’s workbench suite called Programming Development Manager (**PDM**). The Rational Business Developer product includes support for Web development. These two toolsets can be used together to give the developer a rich set of tools. In Chapter 11, we will cover Rational Developer tools for IBM i.
- *Communications support*—This category consists of the many System i Access for Windows (formerly known as Client Access or PC Support) programs that let IBM i connect programmable workstations (PCs) through various PC OSs, including

Microsoft Windows. In Chapter 10, we cover System i Access for Windows in depth and in Chapter 13, IBM i Access for Web.

- *Programming languages*—Traditionally, IBM i systems have supported a number of programming languages by making compilers available as licensed programs. In the past, Basic, FORTRAN, Pascal, and PL/I were supported, but current and future compiler support will be targeted at ILE RPG (RPG IV), C, C++, COBOL, and Java. In addition, Structured Query Language (SQL) is available through the DB2 Query Manager and SQL Development Kit.
- *Web products*—These include IBM i Access for Web, IBM HTTP Server for IBM i OS, and WebSphere Application Server products; and as of V6R1, IBM is shipping Zend PHP and MySQL with the IBM i OS.

Some other licensed program products that do not fall clearly into any of the above categories provide services such as automatic routing, IBM Backup Recovery and Media Services for IBM i OS (BRMS), Performance Tools, and double-byte character support for ideographic languages such as Chinese, Japanese, and Korean. In fact, there are many more such products designed to extend the versatility and usefulness of the IBM i OS.

You can see which licensed programs are installed on your system by going through the Work with Licensed Programs (GO LICPGM) menu.

## User Profile

When we refer to a **user**, we mean any person who is signed on to the system. This may include students, programmers, data-entry personnel, operators, and administrators. To perform work, a user needs to be defined on the system. A user is defined to the system when a security officer or administrator creates a user profile object for that user. A **user profile** not only identifies a user but also describes the user's authority and is the source of several operational characteristics of that user's job.

Later, we examine user profiles in more detail and see that they contain defining information; for example, a user's profile contains the following:

- *User class*, such as casual user, programmer, operator, or security officer.
- *Special authorities*, such as job control, the capability to save system software, or the capability to create or change user profiles.
- *Initial program*, or the program to be run when sign-on is completed.
- *Initial menu*, the first menu the user sees after sign-on.

- *Current library*, the first user library searched for objects the user has requested.
- *Job description*, which contains a specific set of job-related attributes, such as the job queue to use, scheduling priority, routing data, message queue severity, library list, and output information. The attributes determine how a job will run on the system.
- *Group profile*, which assigns a user to a group of users with similar authorities and restrictions.
- *Status*, which indicates whether the user profile is enabled or disabled.
- *Output queue*, or the queue where the user's printer output is stored.

At this time, you need be aware only of the two user-profile values necessary to sign on: *user name* (profile or user ID) and *password*.

## Sign-On

Figure 1.1 shows a normal IBM i **sign-on screen**.

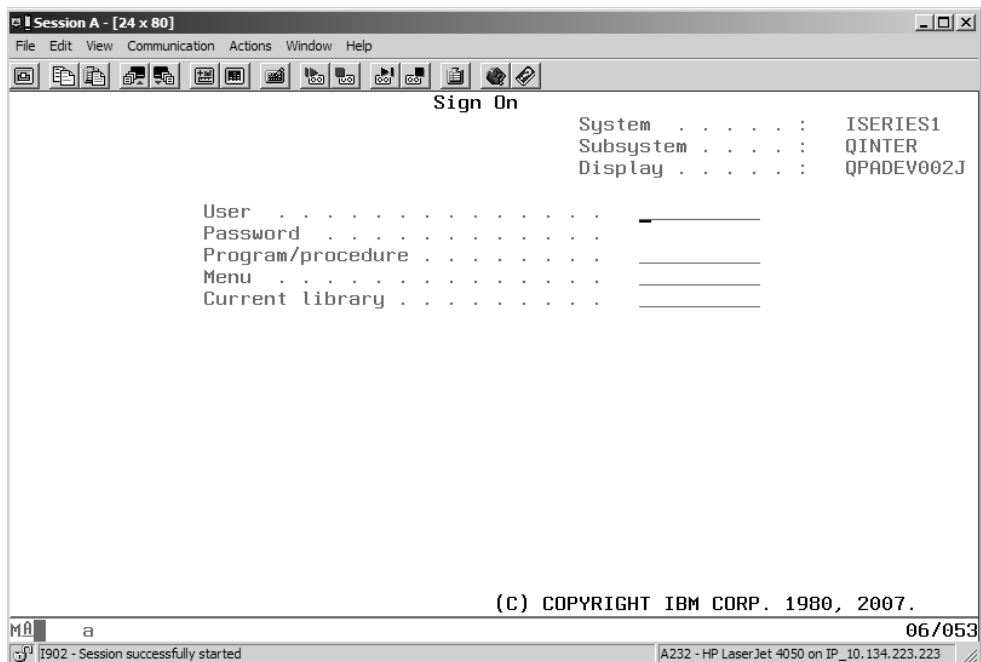


Figure 1.1: Typical Sign-On Screen



The upper-right corner of the sign-on screen displays information that identifies the system, the subsystem the workstation is assigned to, and the display device (or display station).

A **display device** is the workstation hardware (monitor and keyboard) that you use to communicate with the system.

**Note**

As will be the case with many of the figures in this text, certain values in Figure 1.1, such as the System value and the Display device name, will differ for your IBM i system. Although more than half a million IBM i systems are now installed worldwide, it is a safe bet that no two are exactly the same; so screens that deal with configuration values or individual users can certainly vary widely from place to place.

---

To sign on and start an interactive job, a user must type his user profile and password (these values are not case sensitive). A user profile is up to 10 characters long; the security administrator determines its value according to the standards of the installation. The user profile may be a simple first initial and last name combination, such as JSMITH. Or it may be a more symbolic code, perhaps indicating the user's location or department; for example, JMS05KEN could mean a user with the initials JMS of department 05 in Kenosha.

For a sign-on attempt to succeed, the user profile entered must match the name of the user-profile object created by the system security officer or administrator. If a nonexistent user-profile name is input, the system displays an appropriate error message. An error also occurs if the user profile is valid but the password entered at the sign-on screen does not match the current password stored in the corresponding user profile. In either case, the sign-on attempt fails.

When a user enters proper values on the sign-on screen for User and Password, the system proceeds to collect attributes to define the interactive job. Most attributes are taken from a special or default **job description**, but some attributes come from the user profile or from system values.

When sign-on is completed, the interactive job is directed to the subsystem you see on the display (e.g., QINTER in Figure 1.1). Jobs from all physical display devices may run in the same subsystem, or different display devices may direct their jobs to different subsystems. A system administrator knowledgeable in work management and performance tuning makes these decisions.

## User Interface

Unless system defaults are changed, most IBM i interactive jobs begin by displaying the IBM i Main menu, which you see in Figure 1.2.

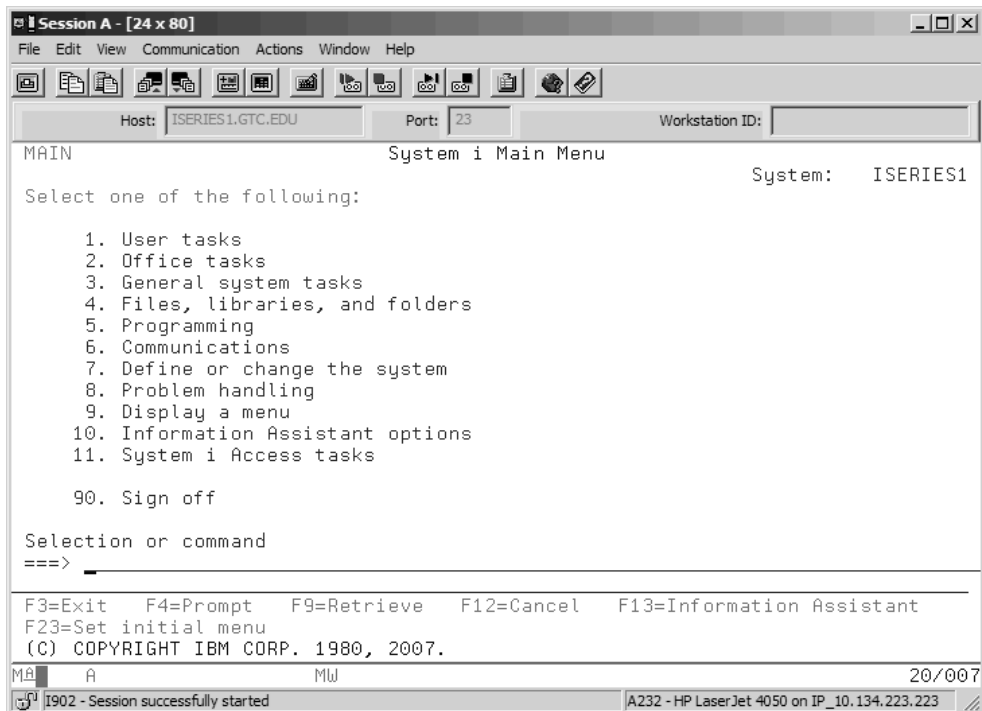


Figure 1.2: IBM i Main Menu

In a hierarchy of menus, the Main menu is the highest-level task-oriented menu, and it can be the starting point to define a menu path to accomplish a specific request. Menus are connected in such a way that a menu choice at a higher level can take you to a lower-level menu. You move through the menu hierarchy until the task is defined to the system. For example, if you wanted to create a library, you would follow the menu path you see in Figure 1.3, starting with the Main menu. We step through this process in the next section.

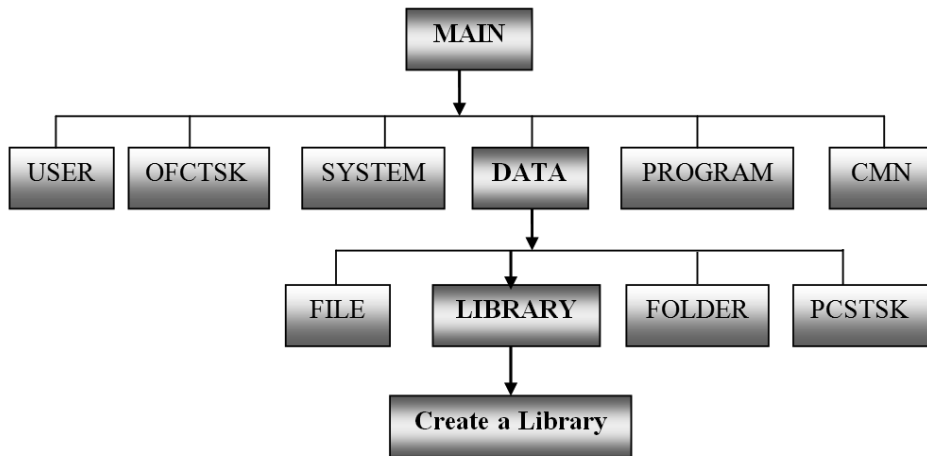


Figure 1.3: IBM i Menu Path for Creating a Library

## Menu Screens

We begin our discussion of IBM i display screens by examining the menu screen format. A **menu screen** has four primary sections, which you can observe if you look again at Figure 1.2:

- The **screen header**, including
  - The menu ID, in the upper-left corner
  - The menu description, centered on the first line
  - The system name, in the upper-right corner
- The numbered list of **menu options**
- The Selection or command line, indicated by `====>` (here, you can type the number of a menu selection or enter a CL command to be executed)
- The list of active **function keys**

Below the two-line list of active function keys is a message line (in the figure, this is the line showing the IBM copyright notice) and a status line that shows cursor coordinates, whether a message is waiting to be displayed, and the status of the keyboard.

All menu screens have this general appearance. The header information and list of menu options, of course, change, depending on the menu. The menu ID in the upper-left corner is the menu object name, the formal name by which the menu is known to the OS. The list of enabled function keys also changes slightly from menu to menu.

Now that you're familiar with the format of a menu screen, let's look again at the Main menu in Figure 1.2. You can take several possible actions from a menu display:

- You can type and enter a menu choice and go on to the next screen.
- You can ask for Help.
- You can type a CL command on the command line and then either prompt for parameters or run the command.
- You can use a function key.

Although most system functions are invoked directly by using CL commands, it is generally easier for novice users to take menu paths to describe the task to perform and then let the system choose the appropriate command. For example, let us use the menu path to create a library—a task you will be performing soon. If you entered a 4 (for Files, libraries, and folders) on the command line of the Main menu and pressed Enter, you would see the DATA menu in Figure 1.4. (Although the description of this menu is Files, Libraries, and Folders, to the OS its proper object name is DATA. You could go directly to this menu by typing GO DATA on the command line.)

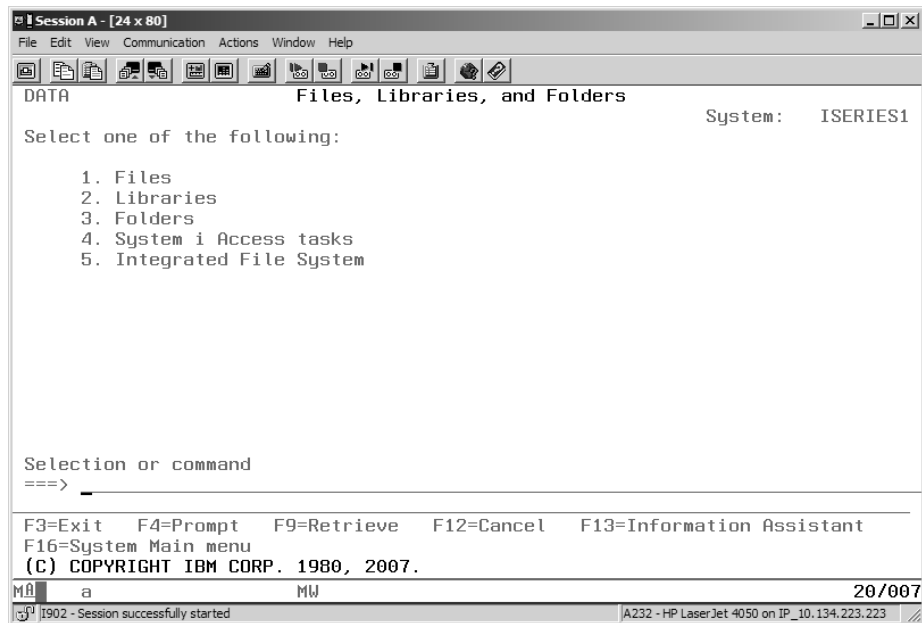


Figure 1.4: Files, Libraries, and Folders (DATA) Menu

Notice the similarity in format between the Main menu and the DATA menu. From the DATA menu, you can select choice 2 (Libraries) by typing a 2 on the command line and then pressing Enter. Doing so takes you to the LIBRARY menu you see in Figure 1.5.

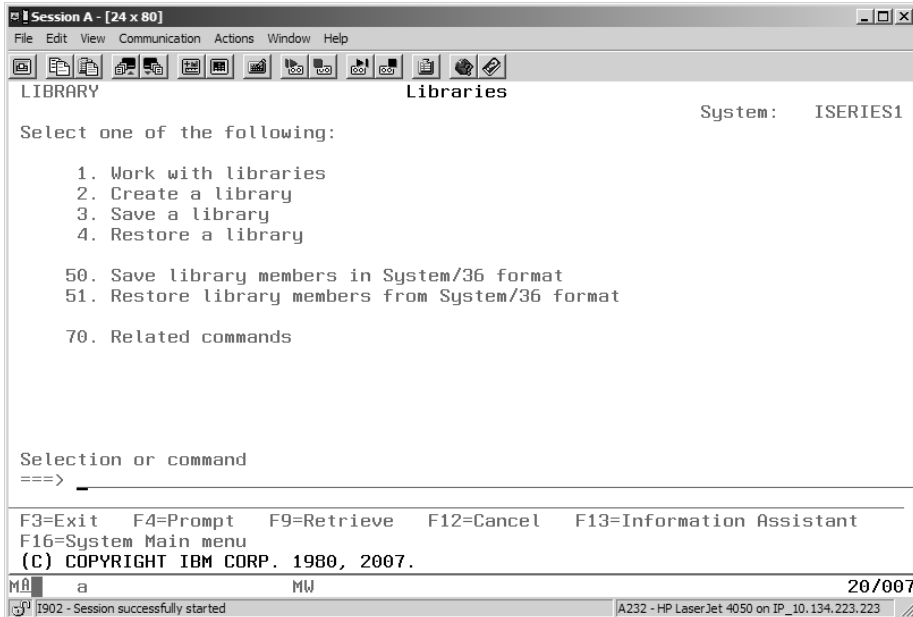


Figure 1.5: Libraries (LIBRARY) Menu

The description of option 2 on the LIBRARY menu, Create a library, suggests that, after three menus, you have just about pinned down what it is you want to do. When you select option 2 from the LIBRARY menu, the system displays the screen in Figure 1.6. The system

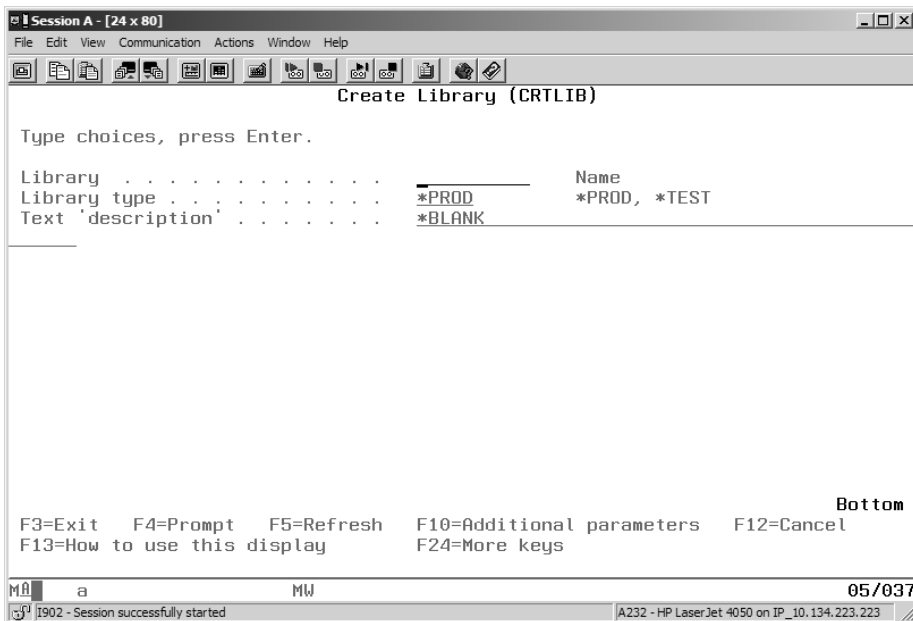


Figure 1.6: Create Library (CRTLIB) Command Prompt Screen

uses this type of screen, generically referred to as a **prompt entry screen**, to request information from a user.

### **Prompt Entry/Command Prompt Screens**

Let us take a moment to study the screen in Figure 1.6. You can see clearly that it differs from a menu screen. First, notice that the heading is different. Rather than the name of a menu, the heading, “Create Library (CRTLIB),” is the description of the CL command in English followed by its IBM i CL command in parentheses. Also in contrast to a menu screen, this screen does not identify the system; and instead of a numbered list of menu choices, you see a list of command parameters. The screen does display a list of active function keys, but no Selection or command line.

This type of screen is considered a prompt entry screen because the system is waiting for you to enter the value or values it needs to process your request. In this case, you enter parameter values for the **CRTLIB** (Create Library) CL command, which the system must execute to create a library. The cursor, as the figure indicates, is positioned at the first character of the entry field for the Library parameter. Because the system is prompting you for a parameter value that it needs to run a command, we refer to such a screen as a **command prompt screen**. A command prompt screen is one kind of entry screen. We reached this particular CRTLIB command prompt screen via a menu path; but if you already know the command name, you could reach the same screen by typing CRTLIB on any command line and pressing the Prompt key, F4.

One option at the bottom of Figure 1.6 is F24=More keys. The **F24 function key** [Shift+F12] will show additional valid function keys for this command, and the **F11 function key** toggles between the display that shows choices and the screen that displays parameter keywords. If you press F24 and then F11, you see that the screen changes to the screen in Figure 1.7. Note that you can request to see all parameters for a command by pressing the **F9 function key**; do so and the screen changes to the one in Figure 1.8, which shows all parameters for the command.

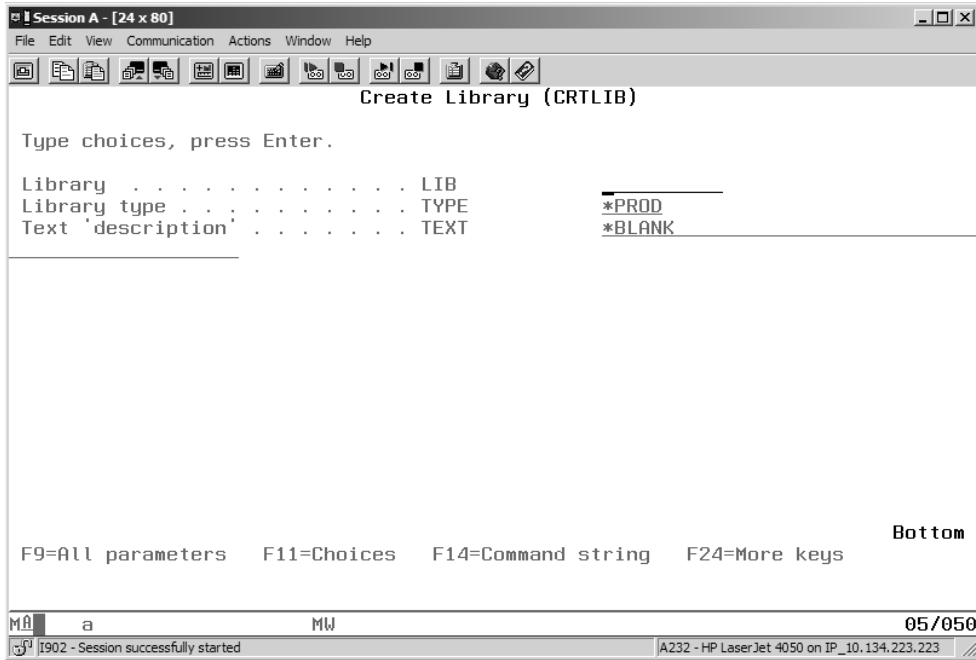


Figure 1.7: Create Library (CRTLIB) Command Prompt Screen, Continued

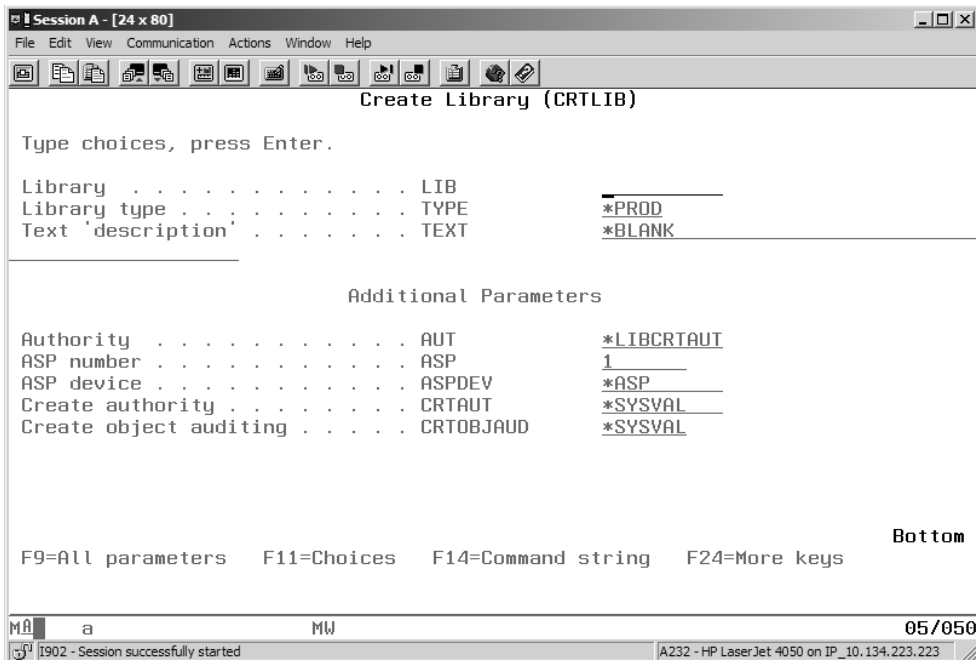


Figure 1.8: Create Library (CRTLIB) Command Prompt Screen with Additional Parameters

If you press F11 to toggle back to the possible-choices screen, you can see the elements that make up a command prompt screen. Figure 1.9 shows the CRTLIB command prompt screen divided into three columns.

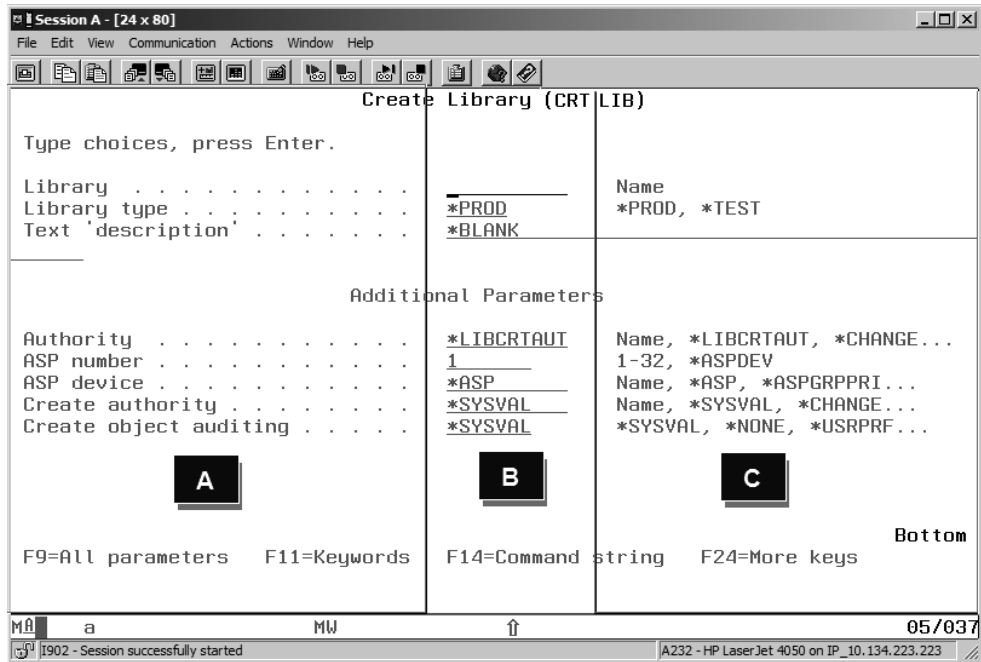


Figure 1.9: Create Library (CRTLIB) Command Prompt Screen Divided into Three Columns

Notice that for each parameter the prompt screen provides a description (A); an entry field showing the parameter's current value, if any (B); and, for most parameters, a list of valid values (C). If the full list of valid values cannot be displayed, an ellipsis (. . .) indicates additional choices. You can view the entire list of valid values by positioning the cursor anywhere on the line that contains the parameter in question and then pressing **F4**, the Prompt function key.

To run a command, you must provide a value for any **required parameters** (i.e., those whose values must be specified for the IBM i OS to execute the command). Required parameters always appear as empty input fields at the top of the command-prompt parameter list. In the case of the CRTLIB command, the Library parameter is a required parameter. An



attempt to run the command without a value for the Library parameter would result in an error. For other parameters in this example, you can use the **default values** provided, or you can type over any default value you need to change. In later chapters we cover command prompting in more detail; for now, it is important only that you can identify a command prompt screen, that you know how to access such a screen, and that you know the reason for using it.

**Note**

The F9 (All parameters) and **F10** (Additional parameters) **function keys** may seem very similar. You can see the difference between the two function keys by keying in **DSPFD** (Display File Description) on the command line and then pressing F4. If you press F10, one additional parameter is displayed. If you press F9, you will see all the parameters for the DSPFD command. Ask your instructor for a file name and library to see the output of this command. We recommend that you use the F9 function key.

---

## Using Help

The third type of screen is an **information screen**. Information screens do not give you menu choices to select from or prompt fields to fill in. They simply provide information that you request and give you a way back to where you were when you asked for the information.

A common type of information screen is the Help screen. You can get Help information from almost any kind of display. Suppose you were creating a library and were not sure what the parameter Library type meant. You could place the cursor anywhere on the line for that parameter (the cursor does not have to be on the input field), and then press the **F1** (Help) **function key**.

### **Field or Context-Sensitive Help**

When you request information about a particular area on a screen, such as the Library type parameter, you are requesting what is referred to as **field Help** or **context-sensitive Help**. To

continue with our example, if you placed the cursor on the Library type field of the CRTLIB command prompt screen and pressed the **F1 Help key**, a window would appear on the screen (Figure 1.10) that contains information about that parameter.

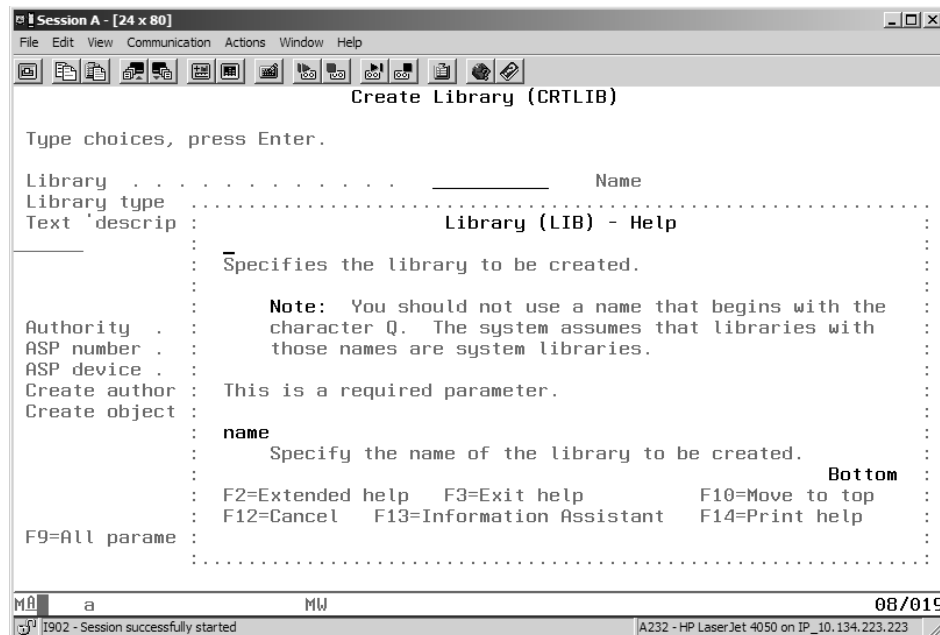


Figure 1.10: Field Help Text for Library Type Parameter of CRTLIB Command

Notice that the window appears just below (or sometimes just above) the field for which you requested Help, and that it has its own list of active function keys. You can enlarge the window, so that it fills up the entire screen, by pressing the **F20 function key**. To cancel the Help request and return to the previous screen, press the **F12 function key**.

### Extended Help

If you were looking at a field Help window and wanted more information about the screen in general, you would press the **F2 function key** for **extended Help**. In our continuing example, pressing F2 from the Library type Help would give you general information about the CRTLIB command. This extended Help would then list and describe all the command's parameters, as well as their possible values. For a screen other than a command prompt screen, extended Help describes the parts of the screen and any entry fields or options available on the screen.

If you press function key F2 (for extended Help) from the field Help screen in Figure 1.10, the system displays the screen you see in Figure 1.11.

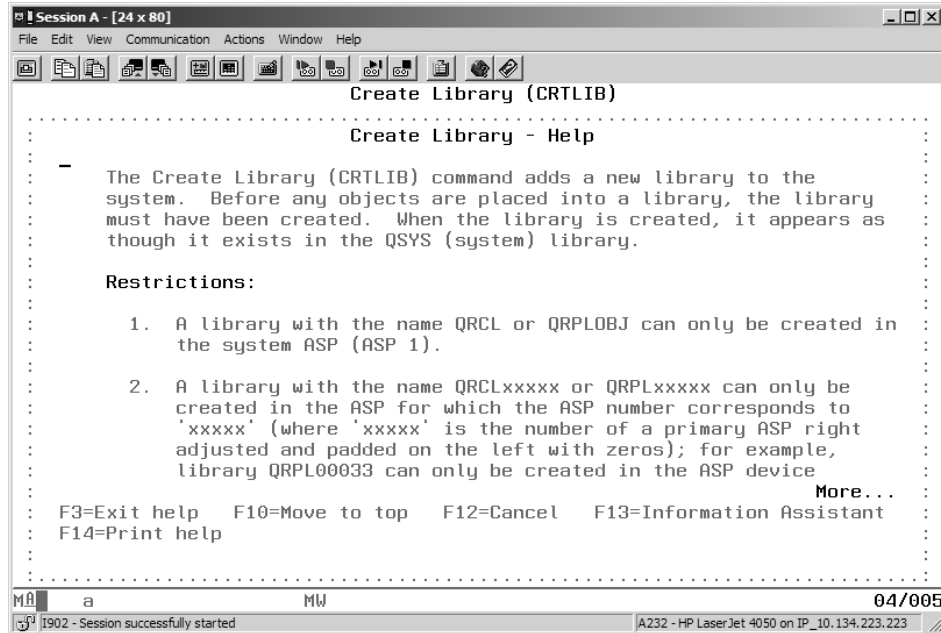


Figure 1.11: Extended Help for the CRTLIB Command

Notice that extended Help begins by explaining what the command (CRTLIB in this case) does. Then it lists and explains any restrictions to the use of the command. Next, it lists the error messages for the command, and, finally, More... in the lower-right corner of the screen tells you that more screens of information are available. To see these additional screens, you would press the Page down (Shift+Roll up) key.

You can get extended Help by pressing F2 from a field Help screen, as we did just now, or you can get it directly by pressing the Help key when the cursor is not in a context-sensitive area of a screen. On a command prompt screen, if the cursor is anywhere on the several lines above the first parameter line—including the screen header—or on a blank line below, pressing F1 should take you directly to extended Help.

The IBM i OS has thousands of Help screens, and you should not hesitate to use them when you have a question about how you can use a certain screen or about the possibilities for responding to a prompt. We rely heavily on Help screens in the lab for this chapter and in subsequent labs.

### **Information Assistant**

Another function key option you will find on Help screens is **F13=Information Assistant**. Pressing F13 takes you to a menu named INFO whose description is Information Assistant Options. (Because this is a menu, you can also access it from any command line by

using the CL GO command—i.e., GO INFO.) The menu has two options, as you can see in Figure 1.12.

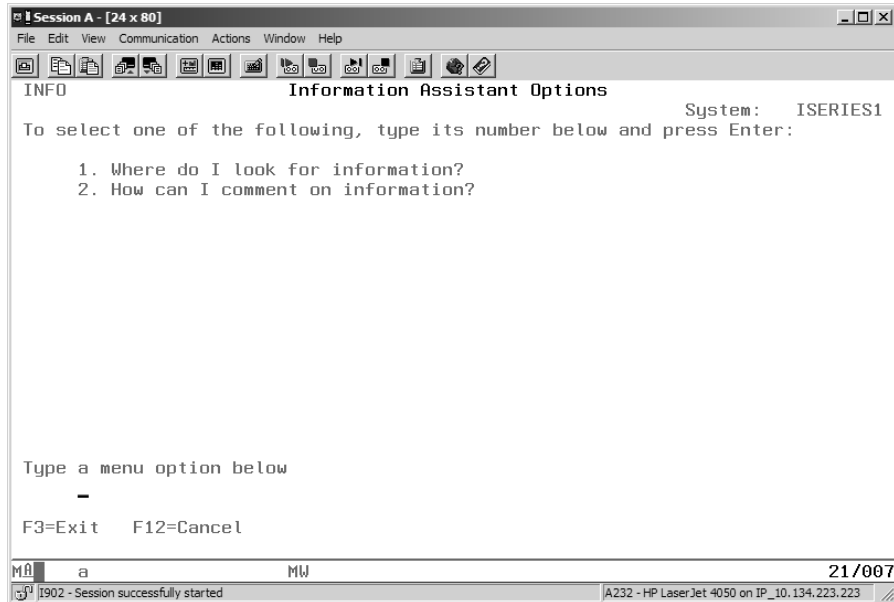


Figure 1.12: Information Assistant Options—F13 from the IBM i Help Screen

Choices 1 and 2 provide a little insight about finding information and providing feedback about the quality of information to IBM. Option 1 displays the screen in Figure 1.13.

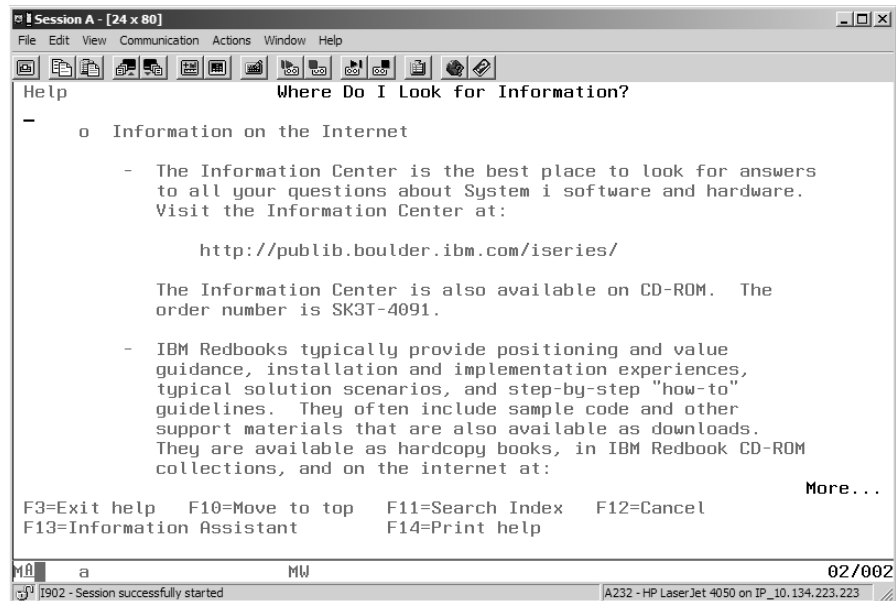


Figure 1.13: Where Do I Look for Information? Screen

In the past few years, IBM has created a Web site called the IBM Information Center. This Web site (<http://publib.boulder.ibm.com/series>) covers every area of information concerning the OS. Notice that in the lower right-hand corner of the screen is More. . . . Pressing the PgDn (Page down) key on your PC will display the screen in Figure 1.14.

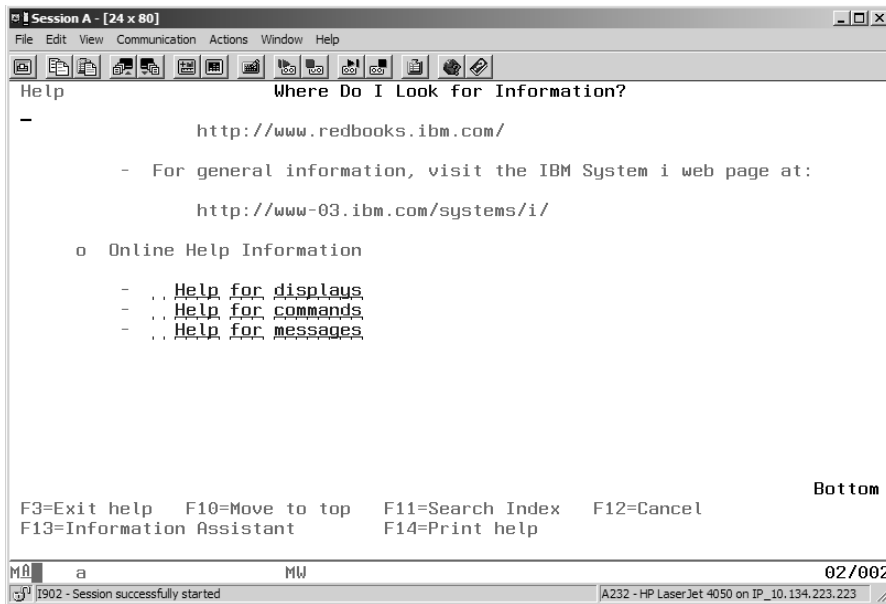


Figure 1.14: Where Do I Look for Information? Screen, Continued

On some Help screens, you see certain words or phrases highlighted with yellow and underlined. Using these **hypertext links**, you may access additional information about these words or phrases by pressing Tab to move the cursor next to the highlighted word or phrase and then pressing Enter.

As you can see from Figure 1.15, the IBM Information Center includes the capability to search for specific topics or keywords. It also allows you to place bookmarks as you search these topic areas and/or keywords. This Web site gives IBM users a centralized location to find the latest information.

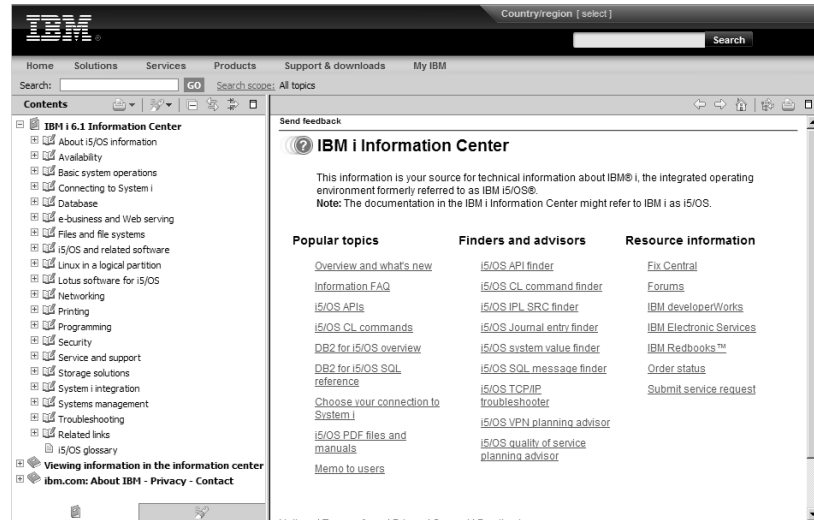


Figure 1.15: IBM Information Center

## List or Work-With Screens

The fourth type of screen that we will discuss is the **list screen**. To demonstrate a list screen, press F12 until you return to the LIBRARY menu represented in Figure 1.16.

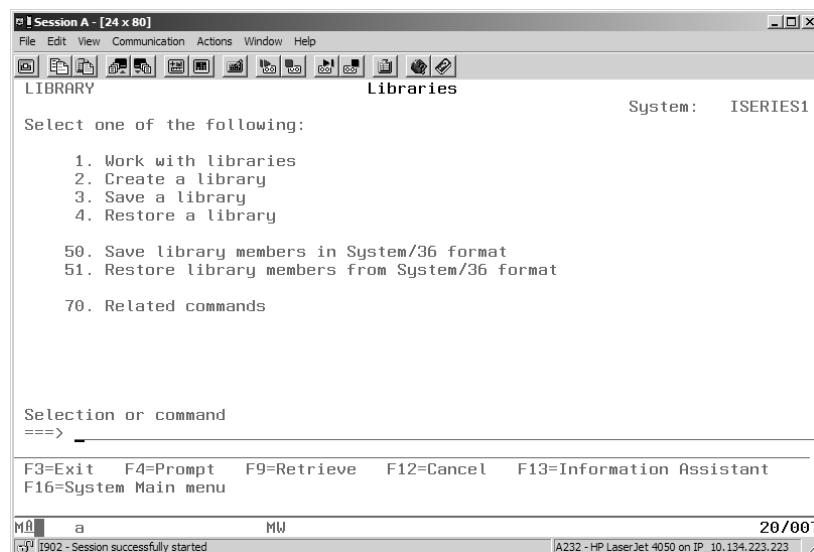


Figure 1.16: Libraries Screen

Select option 1, Work with libraries, and press Enter. Doing this will display the screen in Figure 1.17.

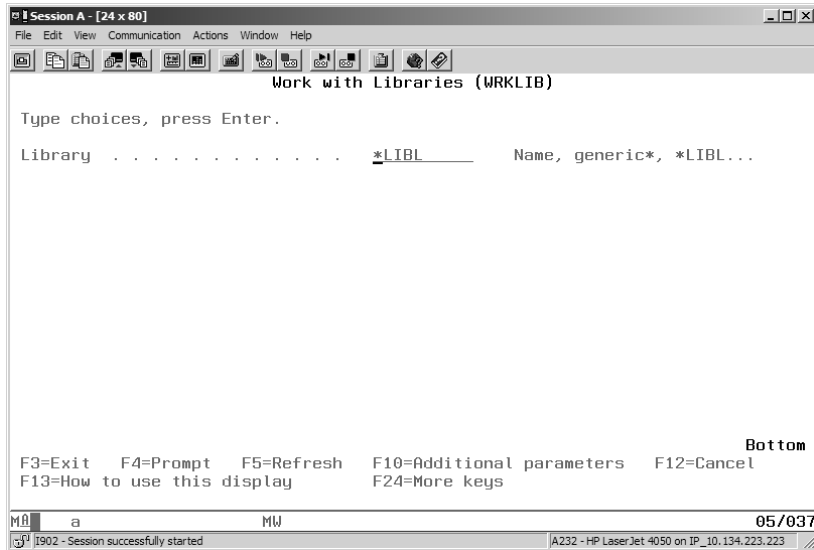


Figure 1.17: Work with Libraries (WRKLIB) Screen

This screen allows you to key in a parameter to control what libraries will be displayed. The default is \*LIBL, which means that the libraries in your library list will be displayed. Notice that the screen name includes both the Work with Libraries menu name and the related CL command (WRKLIB). After you press Enter, the system will display Figure 1.18.

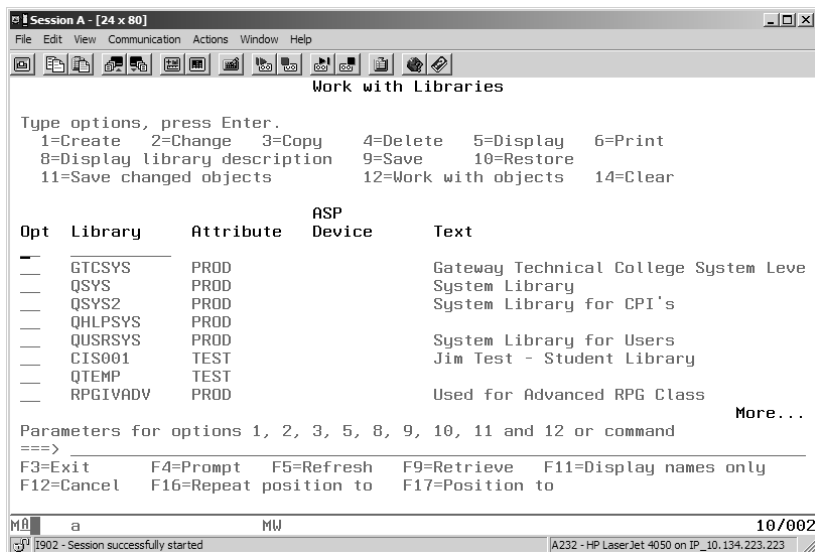


Figure 1.18: Work with Libraries (WRKLIB) Screen, Continued

The type of screen you see in Figure 1.18 is called a list screen, a list-processing screen, or, often, a “work-with” list screen because it so commonly results from running a “work with” CL command. We use these list screens extensively in our labs and explain more about their power and flexibility in later chapters. You also could have typed this command in the command line to access this screen.

Note that the first line of text under the screen heading in Figure 1.18 says Type options, press Enter. Some of the options provided are

- 1=Create
- 2=Change
- 3=Copy
- 4>Delete

The number of options on a list screen is determined by the command that produced the screen. You should always understand the ramifications of using these options.

We now have briefly discussed four different types of screens that IBM i systems use to communicate with interactive users. They are

- Menu screens.
- Entry screens—Command prompt screens are a common type of entry screen.
- Information screens—Help screens are frequently used types of information screens. Help screens include extended Help and field or context-sensitive Help.
- List screens, also known as list-processing or work-with list screens.

Being familiar with these screen types and knowing how each is used can help you overcome that sense of being lost in a foreign environment that users so often have as they learn about a new system. As you work through the lab exercises, try to identify each screen you encounter—doing so will help you maintain your orientation. Think about how you got to each new screen; and if you do get lost, remember that pressing F12 backs you out one screen at a time. When you successfully complete a task, use F3 to return to a common starting point, such as the Main menu.

## **In Summary**

In this chapter, we examined some important IBM i OS characteristics. We discussed the interactive nature of the system and the idea of how jobs are run in subsystems. We know that an interactive job is started when a user signs on to the system, and that, at normal operational security levels, a user profile must exist for a sign-on to be successful.



Once on the system, most of us rely on several different types of screens to accomplish our tasks. As novice users, we do most of our work on the system by traversing a series of menus. As we progress down the menu path, we are clarifying to the system what task we want to accomplish until an appropriate command can be executed. The command may take us to another screen, such as an information screen, or it may run a program; or, if the command has no required parameters, it may invoke a system function.

We also examined the IBM i's Help facility. We determined that for most system displays, Help is available both at the extended and at the field or "context" level.

As you work through subsequent lab exercises, you should always try to maintain your orientation—what kind of display you are at, what the purpose of that display is, and how you got there. Use Help freely when you have questions about a display or are uncertain about the meaning of an input field or the use of a function key.

One of the outstanding features of the OS is its comprehensive Help system. If you need Help, remember F1, and then F2.

## Key Terms

batch jobs	F12 function key
CL	F13 function key
CL program	F20 function key
command line	F24 function key
command parameters	function keys
command prompt screen	Help key
context-sensitive Help	HLL
CRTLIB	hypertext links
DASD (or hard disk)	IBM i OS
default values	Information Assistant
DFU	information screen
display device	IPL
DSPFD	interactive job
extended Help	job
field Help	job description
F1 function key	job queue
F2 function key	keyboard mapping
F4 function key	library
F9 function key	library list
F10 function key	licensed program products
F11 function key	list screen

menu interface	SDA
menu options	sign-on screen
menu screen	SEU
multitasking	subsystem description
multiuser	subsystems
object	system values
object type	QCTL
PDM	QINTER
prompt entry screen	QBATCH
RDP	user
required parameters	user profile
screen header	work management

### **Review Questions**

1. What are the two types of jobs discussed in this chapter?
2. Explain the function of the two job types from question 1.
3. List and define the different categories of licensed programs.
4. Explain the term “object” in the IBM i OS.
5. What is the purpose of Control Language (CL)?
6. List and describe the categories of system values.
7. Using the online IBM Information Center, list and define three licensed programs not discussed in this chapter.
8. What is the maximum character length of a user profile?
9. Are the user profile and password case sensitive?
10. What is a user profile?
11. List and define the categories of a user profile discussed in the text.
12. List and define the four primary sections of a menu screen.
13. What does pressing F20 accomplish in a Help screen?
14. How do you prompt a command?
15. When prompting a command, how would you display all the parameters for the command?

## Lab 1

### Introduction

This lab is intended to provide experience with the main topics we covered in the Chapter 1 text: system values, licensed program products, and basic CL command use. We examine these topics through the use of the four types of screen displays: menus, information screens, entry screens, and lists. Rather than building objects or generating output, the goal of this lab is for you to become accustomed to recognizing and using the four types of screens by having you use them to explore some important system components.

### Part 1

**Goals** Recognize the IBM i OS sign-on screen.

Sign on successfully.

**Start at** IBM i sign-on screen.

**Procedure** Enter a valid user ID and password.

- 1.1. You should see the IBM i sign-on screen, similar to Figure 1.1 (page 8). There are many different possibilities for connecting workstations to a server running the IBM i OS; your instructor/mentor should provide you with specific instructions. If you need additional help, ask your instructor/mentor for assistance.
- 1.2. The sign-on screen is a special type of entry screen. You are expected to key in and enter certain information to identify yourself to the system, but function keys and Help information are not available.

Look at the identification information in the upper-right corner of the screen.

- 1.2.a. What values are shown for the System, Subsystem, and Display attributes of your workstation?

The Display name is the name by which your workstation is known to the system. It will also be the name of your interactive job.



#### Note

Before you sign on, let us warn you about **keyboard mapping**. Keyboard mapping is another area in which differences exist from one installation to another (or even among different types of keyboards or connections within installations). In general, the Tab key on a PC keyboard equates to the 5250 (older, nonprogrammable workstation) keyboard's Field advance key. The 5250 Field exit key, which erases everything in the current field from the cursor location and then jumps to the next input field, may be mapped on PC

keyboards to either the Enter key or the right Ctrl (Control) key, depending on which 5250 emulation product was used (or whether remapping was done). Other products, such as the popular MochaSoft TN5250 (which sells versions for Windows and Mac), map the Field exit key to the large “+” key on the numeric keypad.

There are too many mapping variations to try to list all of them here; but if you’re working through a Windows client, you should see a toolbar with a keyboard button at the top of your screen (or you can try clicking Windows Help for “Keyboard Mapping” information). Clicking the keyboard button lets you find out how various system functions are mapped on your keyboard.

---

Begin the sign-on procedure by typing the user profile that you were assigned by your instructor. For example FLastname, CIS001D, or JMS05KEN.

Your user profile will appear in capital letters, even if your keyboard is not shifted to upper case. (Don’t worry about case.) Remember, a space character is not a valid character within a name. A user profile has a maximum length of 10 characters. If you have problems, ask your instructor/mentor for help.

Don’t press Enter yet. If your user ID is 10 characters long, the cursor will already be in the Password input field; otherwise, use the Tab, New line, or Field exit key to move there:

Tab (or New line or Field exit)

Next, type the password that your instructor gave you. The password is not case sensitive, and for security purposes the characters are not displayed on the screen; so type carefully.

If you think you made a mistake and you haven’t pressed Enter yet, backspace and type over what you’ve entered. If you typed too many characters the first time, press Field exit (after you’ve typed your password) to erase the rest of the input field above and to the right of the cursor.

Once you’ve keyed in your password, press Enter. (Remember that the Enter function may be mapped to the right Ctrl key on your PC keyboard.)

Enter

If an error code and the message “*Password not correct for user profile*” appear near the bottom of the screen, try rekeying your password and then pressing Field exit and Enter. If the message persists, ask your instructor/mentor for assistance.