

IBC233 Frequently Asked Questions and Other Problems

I am stuck. The System i gives me a message at the bottom of the screen. What do I do?

It may just be a keyboard error, make sure Scroll Lock is off and press the ESC key.

... It's not that.

Put the cursor on the message and press F1 to find out more information. Sometimes, the difficulty lies not in the step you are trying to do now but in a previous step. Check the previous steps in the lab or assignment for correctness.

What was that message again?

The DSPJOBLOG (Display Job Log) command can be very useful if a system message has gone away and you want to look at it again. The job log is deleted when you sign off and your job ends normally. The job log is retained if your job ends abnormally.

The PDM "3-Copy" option does not work.

Remember, the System i is not a PC -- everything does work all the time.

PDM object Option 3 does *not* run the Copy *file* command. Specify option 3 and use F4 (instead of Enter) to find out what option 3 does. Try it out. Use Help on the error message.

Then, use F23 to find the Copy *file* option. It works fine.

PDM member option 3-Copy *does* work when copying a source *member*. It will not work for you when trying to copy an *object* from the course library.

There is a Date data type field on my screen by my CL compile says:

CPI0301 10 Character CL variable declared for date field &xxxxxxx.

Control language (CL) does not support Date type variables, so the variable is declared as a character variable by the compiler. This is a severity 10 message -- it is just a warning to let you know that the compiler changed the data type from what the *FILE object specified. See below for compiler messages.

What does the CPF4131 "level check" message mean?

It means you changed your *FILE object but did not recompile the program which uses that object. The file is now at a different level from what the program expects -- data corruption would occur. Recompile your program and check that it compiled successfully.

What does the CPF0001 "received by program at line-number." message mean?

It means the system could not run a command in your CL program. This is not very helpful. However, if you *put the cursor on the CPF0001 message and press F1 to find out more information*, you will notice that function key "F10=Display messages in job log" is available. Press F10. Listed above the CPF0001 messages will be the reason the error was issued. Put the cursor on each of those messages and press F1 to find out more. If you miss this step, you can always run the DSPJOBLOG command yourself later.

Under no circumstances should you pay attention to the system's Recovery suggestion: This inquiry message can be avoided by changing the program. Monitor for the error (MONMSG command)...

That recovery suggestion can apply to some CPFnnnn messages but definitely not to this one. CPF0001 is a symptom message. You must find the cause. Your objective is not to avoid messages but to correct the reason they are issued.

I cannot compile anything or create any printed output

Check the job log for messages ==> DSPJOBLOG

Check that you still have an Output Queue object (*OUTQ) in your default library. Both the library and the *OUTQ have the same name as your User ID.

If you accidentally deleted the *OUTQ object, recreate it

==> CRTOUTQ Dx233snn

Other Problems: Writing, Compiling, and notes on life's little problems

[Editing and Compiling](#) [Debugging](#) [Problem Solving](#)
[System i Sign On problems](#) [Other Problems](#)
[Ohnoitsallgone](#) [FAQ](#)

The ~~GO-SENECA~~ menu has many useful things to help you use and solve problems on the System i.

Editing and Compiling ^

Syntax errors

- SEU will find most simple errors during editing
- use F4 to prompt the command or statement — the system will assist you with parameters, syntax, and provide help.

The system sends you a message whether your compile ran successfully or ended abnormally. It also creates a compile listing (a spooled file).

If it ended abnormally, a *PGM object is not created or the previous *PGM object is not replaced. The compile listing will tell you what you need to know.

If it was successful, a *PGM object was created. This does not mean you can ignore the compile listing — check for warning messages.

CLP compiler diagnostic messages look like:

```
Msg id  Sv  Message text
* CPD0789 00  No command specified on IF or ELSE command.
* CPD0791 00  No labels used in program.
* CPD0084 30  'CHAR      ' not valid for parameter TYPE.
* CPD0727 40  Variable '&MSGTEXT  ' is referred to but not declared.
```

Sv = severity meaning:

- 00 or 10 is a warning. Sometimes you can ignore them (e.g. CPD0791 above). Sometimes you cannot (e.g. CPD0789 above).
- 20 is a serious warning which *may* prevent a successful compile
- 30 or higher is an error which *will* prevent a successful compile

Compile errors ^

- The system will send a message telling you if your compile was successful or not. The PDM user option, DM (DSPMSG), shows your messages. If the compile was successful, you should still review the compile listing for warning messages.
- The PDM user option, SP (WRKSPLF), shows your compile listing(s) and, if there are errors, a job log. However, there is an easier way to review a compile ...

The next time a compile fails, enlist SEU's editor to help you find the errors.

- start SEU in edit mode and
- press **F15** for Browse/Copy options, then
- select option **2** to access the compiler spool file.
- the SEU panel splits into two halves. The upper half still shows the source member in edit mode, while the bottom half shows the compiler listing in browse mode.
- Move the cursor to the SEU command line in the bottom half, and execute the F (find) command to locate the first error:
- SEU==> **F *ERR**
- SEU displays the first line in the compiler listing with an error, and displays the error message at the bottom of the screen. Move the cursor to the message line and press Help (**F1**) to get cause and recovery information -- this is the second level text optionally printed in a compile listing.
- To fix the problem that caused the error message, move the cursor to the upper half of the screen and make your corrections. Then, move the cursor to the bottom half and press **F16** to find the next compiler warning or error message. Continue this process until you reach the last one. F12 takes you back to full-screen edit mode.
- This technique works with all types of source code, not just with CLP. You can use it just as easily with DDS, and COBOL or RPG programs.

Debugging ^

Run Time Errors

- Put the cursor on the error msg line
- Press F1
- Read the message and note the program's line number and other specific information.
- You may have to examine the job log for preceding messages that further explain the problem. Use F10.

Data Errors

- if your code looks right, perhaps the problem is with the data.
- look at the program's output to determine the logic or data problem ...
- if nothing was output, perhaps the error occurred on the first read
- if everything was output, perhaps the error occurred after the last read
- if the same data was output many times, you are not processing all the input records
- check your library list
- use the Work File command, e.g. WRKF filename*. This cmd lists files across your library list. PDM shows you objects only one library at a time.
- use RUNQRY *N filename and display the data you think your program should be reading

Logic Errors

- you did check your compile listing for warning messages, didn't you?
 - use the debug utility, Start the [Interactive Source DeBug](#) utility (STRISDB), and walk through your code
-

Problem Solving ^

1. RTFM (Read The *Fine* Manual). About half the questions I get are already answered in the lectures, textbook, on my web pages, in the lab or assignment specifications, and in examples in the course library's source files. If you have not read this material at least TWICE, you have not really read it. There is a good chance you can solve the problem yourself in minutes just by rereading the material.
2. The System i will help if you put the cursor on the msg and press F1.
3. When fixing compile errors, use SEU's browse function and "F *ERR" command to find errors and use F1 on the diagnostic message. (see above)
4. Use DSPJOBLOG to investigate run time errors. A significant part of what I know about the System i was learned just by doing this. (see above)
5. leave the problem alone for a while and come back later. A good night's sleep is often the best help you can give yourself. After midnight, problems can take hours to solve. In the morning, they usually take minutes.
6. show a colleague your problem. Another pair of eyes can often spot the problem. Even experienced programmers sometimes need someone else to look at their code. Also, explaining things out loud to someone else can often reveal the problem's cause.
7. take the CURRENT COMPILE listing of your program to the Learning Commons Help Desk or to one of our tutors. There are people there who have taken this course before who can help.
8. when testing your logic, make one change at a time and re-test. Making too many changes at once can complicate debugging. When you are all done, re-test everything to make sure a fix did not "unfix" something else.
9. PRINT the CURRENT COMPILE listing of your program (and screens) and bring it to me during office hours with the problem code/messages highlighted.
10. send your instructor an email. In your email:
 - cut and paste the error message(s) from the compile or screen,
 - cut and paste the line(s) of code in error and enough lines before/after those lines to give some context
 - DO NOT COPY A GRAPHICS IMAGE OF THE SCREEN. just copy the text only.
 - say what you did to try to fix it. What does not make sense to you? Why do you think it should work (even though the system has a different opinion)?
 - Be careful about your email program automatically wrapping long lines when sending the msg -- do not allow it to do this -- it makes reading code very difficult.

"Please tell me why my program will not compile/work" is NOT a good way to ask for help. Instructors do not debug students' programming problems. That is your job. Their job is to help you do your job.

System i Sign On problems ^

If you have difficulty signing on, you have one of two possible problems.

PROBLEM 1:

If you can sign on and see this screen:

```
+-----+
| Previous sign-on..... mm/dd/yy |
|                                     |
+-----+
```

you think everything is normal but then see a screen that looks like this:

```
+-----+
| Job: QOADEV005L User: DC234Z99 Number : 123456 |
|                                               |
| Job ended abnormally.                       |
|                                               |
| Display the job log for more information.    |
|                                               |
| Press enter to continue.                    |
+-----+
```

and when you do press Enter, the system returns you to the sign on screen. You have a problem in your user profile.

Don't worry, you can fix the problem yourself by signing on with overrides, then correcting your profile.

To sign on with overrides, specify the following on the sign on screen:

```
+-----+
| User . . . . . your user id |
| Password . . . . . your password |
| Program/procedure. . . *none      | <== override
| Menu . . . . . main              | <== override
| Current library. . . . qqpl       | <== override
+-----+
```

After you have signed on, examine the spooled file containing your last job log – it will tell you what the problem was. Prompt the CHGPRF command with F4 and change your profile settings as needed. Then sign off and try to sign on again normally.

PROBLEM 2:

I still cannot sign on!

If you see this message displayed at the bottom of the sign on screen:

CPF1394 User profile Dxxxxxxx cannot sign on

Please see your instructor who can reset your profile or contact ACS.

The system will automatically disable your profile after too many unsuccessful sign on attempts (to guard against password cracking).

It could also be because an administrator disabled your profile when you were running too many infinite loop jobs.

When you can sign on again, please see the SENECA menu item that explains how to cancel a program in an infinite loop: ==> GO SENECA

Other Problems ^

By the way, NEVER give someone else your password so they can look at your code in your library -- not even a professor. It breaks our [computer use guidelines](#).

If any of these things happen to you:

- "error found on STRSEU command" when you try to edit a member
- cannot compile anything
- no compile listing is found after compiling something
- you change your spooled file to PRT01 but it just sits in the queue

then you accidentally deleted your output queue (see above) or your profile has probably used up its maximum space allocation.

Most of the space is likely occupied by old compile objects kept in a library called QRPLOBJ (replaced objects). The system keeps these because of the high security level we use on this System i. QRPLOBJ is cleaned out at each IPL but if you do a lot of compiles, your space limit may be exceeded before then. (IPL = Initial Program Load = reboot)

The best way to clean up this junk is by running the WRKOBJOWN command and paging down to the stuff in QRPLOBJ library (F18 may quickly get you to these objects). Delete ONLY the objects in QRPLOBJ. Most of these objects begin with Q and have a cryptic name. *Be careful not to delete your good programs in your own libraries.*

After this clean up operation, you should have enough space on the System i for the temporary work files needed by SEU, compiles, printing, and your own new programs and files.

Ohmygoditsallgone ^

It happens every semester. Someone thinks they were "cleaning up" junk when they were really "cleaning out" their work.

DO NOT "clean up" anything until *after*...

- a good night's sleep.
- you have [FTP'd](#) all your source code from every one of your libraries to a diskette/USB drive.
- you have made a handwritten list of where the good stuff and the junk is located.
- double checked everything with a partner.
- you are sure what objects you are working with (a member list can look at lot like an *OUTQ if you have been working too long and too hard which is undoubtedly the case).

It is a good idea to [FTP](#) your code at least once a day and have at least two different diskettes (in rotation) to save it on.

When [using FTP](#), be sure to use ASCII transfer (not Binary or Auto)

Remember, the ~~GO-SENECA~~ menu has many useful things to help you use the System i.

[The HTML version of this document](#)