BCI433 Lab 4 CLLE Interactive Program (updated Winter 2020)

Lab Objectives

- **Create Filters**
- **Code a Client Access sign on program and understand User Profiles**
- **Retrieve system values, job attributes and user profile**
- **Create and use a startup program**
- **Monitor for Error Messages**

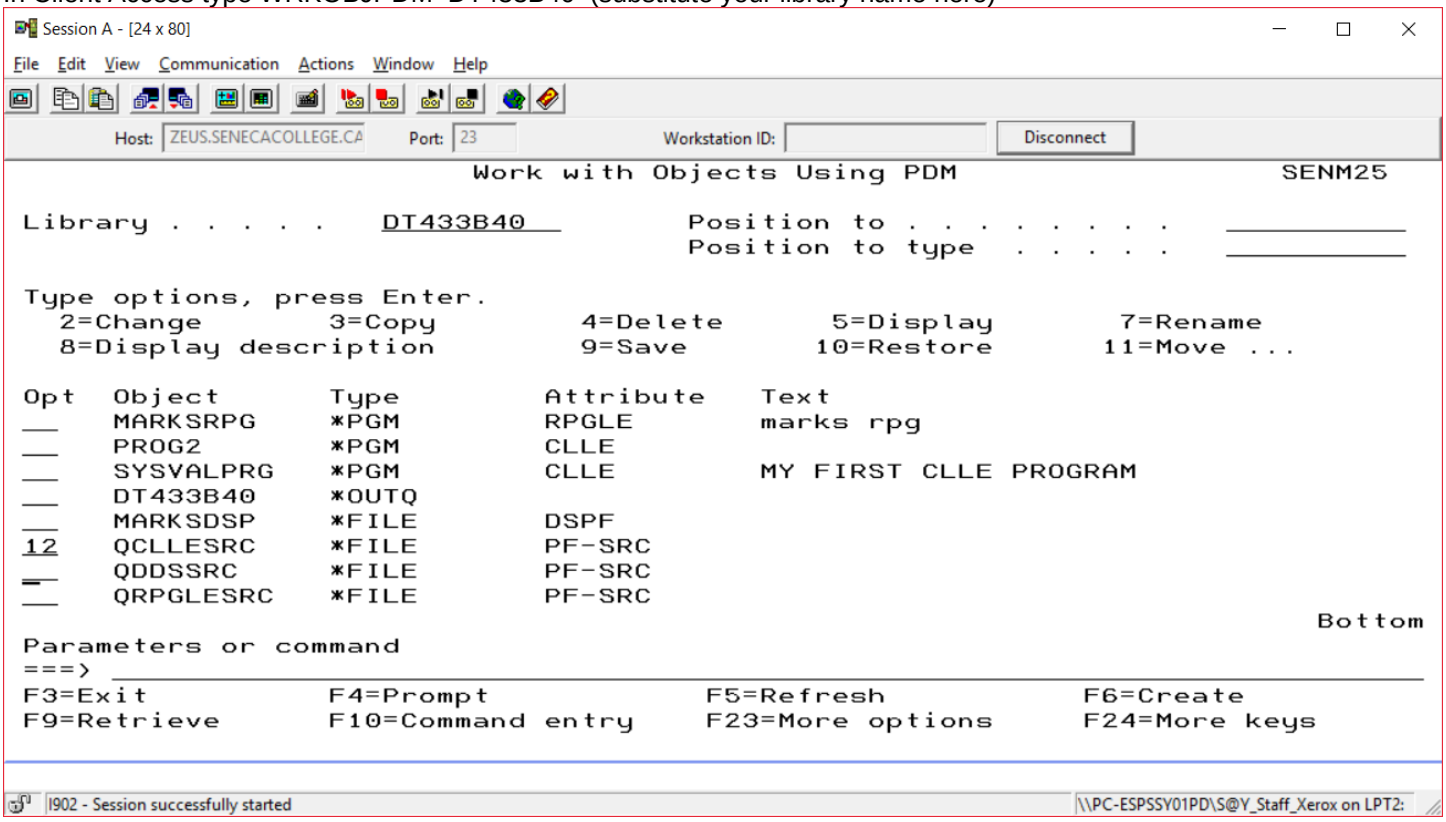
Lab Requirements Lab 4A (next week) Lab4B (in two weeks)

- **Startup program runs on Client Access sign on (4A)**
- **Demonstrate Lab2 and Lab3 filters (4A)**
- **Demonstrate the Lab 4 CLLE program with only validation working (4A)**
- **Demonstrate a filter command (4A)**
- **Demonstrate the Lab 4 CLLE program with everything working (4B)**

Part A – Working With Objects

Sign on to a Client Access and an RDi session.

In Client Access type WRKOBJPDM DT433B40 (substitute your library name here)



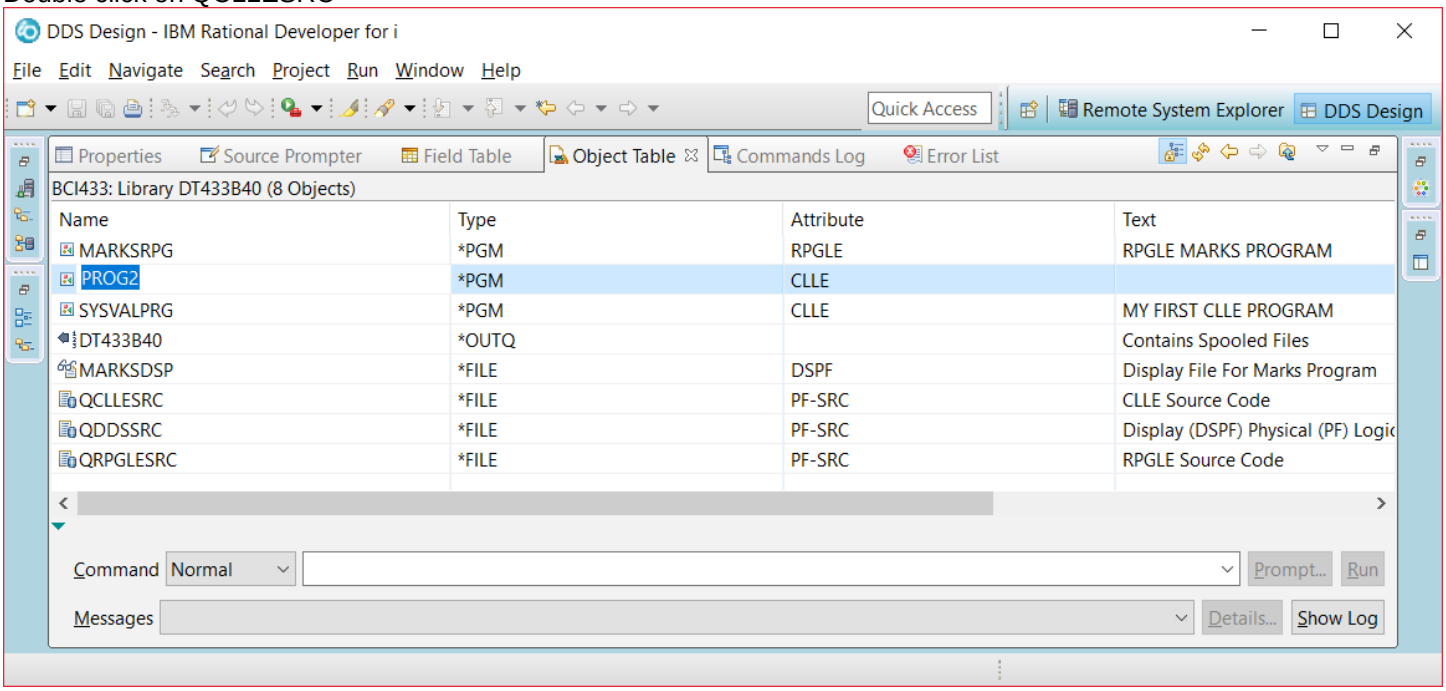
You can look at the QDDSSRC program members in PDM by typing a 12 beside the name and pressing enter.

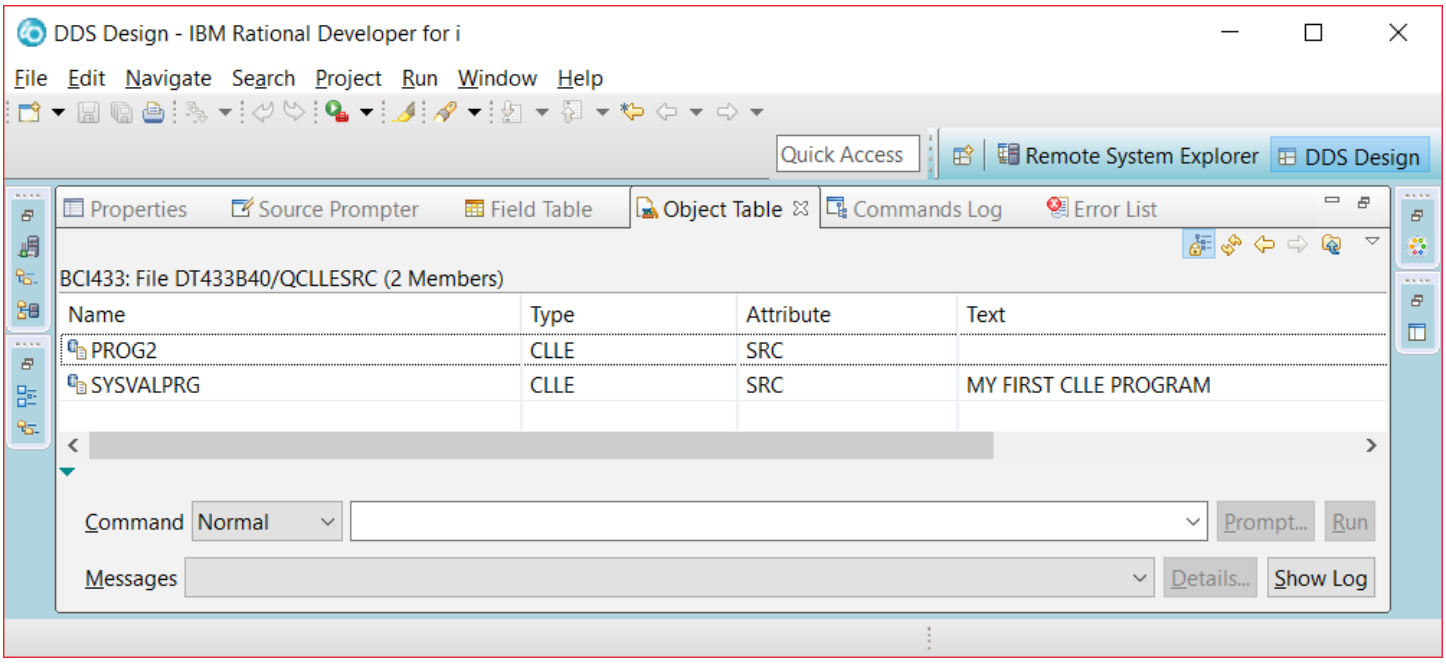
The same thing can be done in RDi

First right click on your library name and select the **Show In Table** option. Then double click on the Object Table tab to provide a full screen of object names, types, attributes and text comments.

Note: LAB 3 is changed some semesters. So you will see different RPGLE and DSPF members.

Double click on QCLLESRC

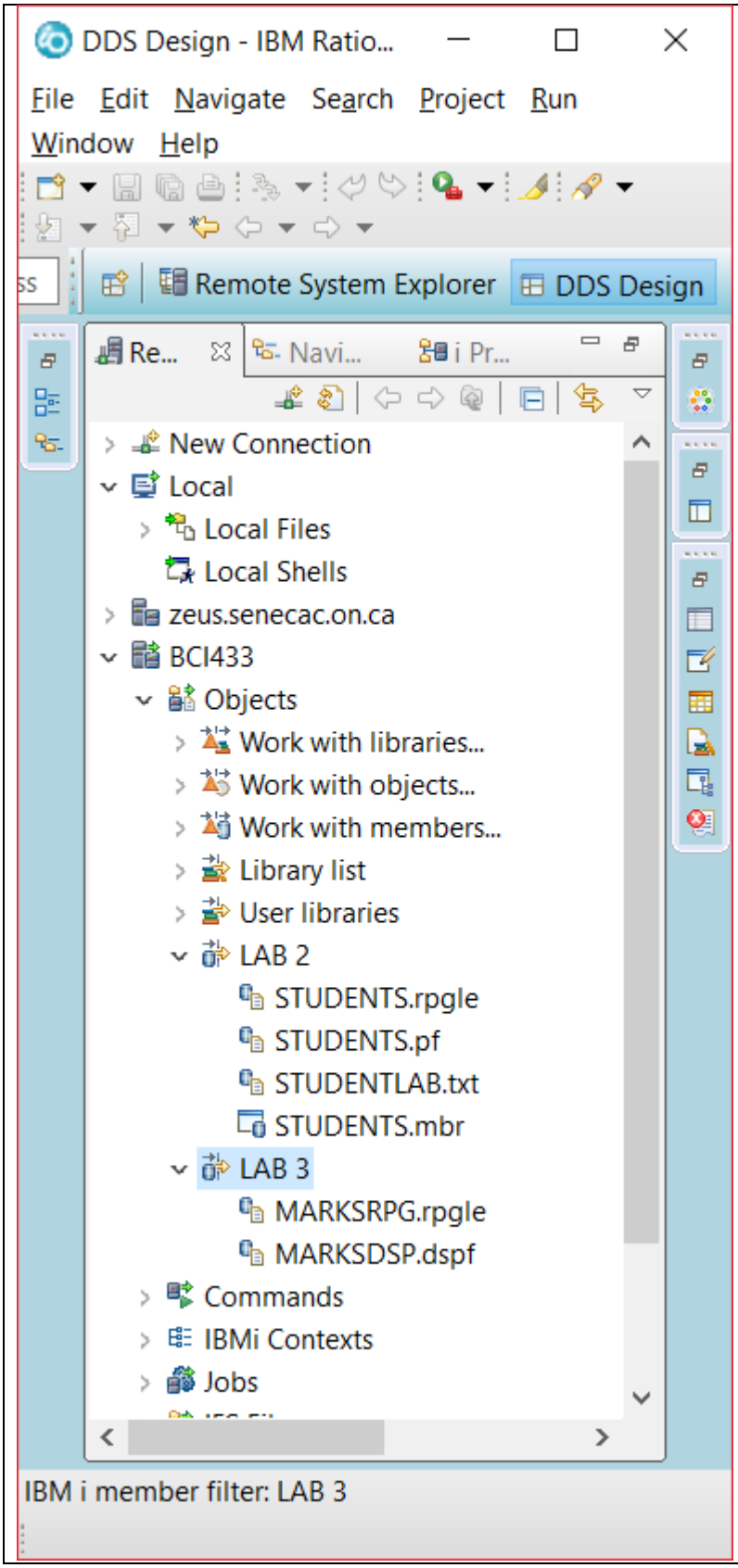




Click on the arrow showing below Remote Systems on the top left to go back to the objects view

Part B – Filter by Application

The following Lab2 and Lab3 RDi filters were demonstrated in class. Provide them in your workspace and be prepared to demonstrate them.



In order to collect objects and members by application rather than member type do the following:

Under Objects

Work with members
Provide Library, File, Member name
Provide Filter name

Right Click Created Filter

Select Properties
Click on New Filter String
Provide Library, File, Member name

Repeat as necessary

Lab 2 Filter

STUDENTS.rpgle in QRPGLESRC in your library
STUDENTS.pf in QDDSSRC in your library
STUDENTLAB.txt in BCI433LIB/StudentLab
STUDENTLAB member
STUDENTS.mbr contains data allowing
use of Show in Table

Lab 3 Filter

This shows from as MARKSRPG and MARKSDSP. The lab3 application changes each semester but there is a naming convention. So you will have a ?RPG and a ?DSP.

Hint: - When adding members from your library, do not browse library or file, just browse members – it will save you time.

Part C – Write the Startup program – MONMSG Command.

Write a simple CL program called **STARTUP** that does the following:

1. execute the program STRJOB in the library QGPL.
2. Add BCI433LIB to the user portion of your library list.
3. Display the system level (DSPSYSLVL). Send the output to *DISPLAY.

After successfully compiling your STARTUP program, start a 'Green Screen' (emulator) session. (Client Access Session). Look at your library list before running the STARTUP program and then look at your library list after running the startup program.

What is different in the user portion of the library list? It would have the BCI433LIB included

Call your new program, STARTUP again.

What you see is affectionately called the 'Black Screen of Death'! Your program has crashed. Unlike Microsoft's Blue Screen of Death, IBM's BSD gives you a lot of information about why and where your program crashed – you just need to look!

```
                          Display Program Messages

Job 659632/DY433D40/QPADEV000B started on 05/07/07 at 09:13:37 in subsystem QI
CPF2103 received by procedure STARTUP. (C D I R)

Type reply, press Enter.
  Reply . . .     _____________________________________________________

F3=Exit    F12=Cancel
```

The first line of the Black Screen of Death tells us what job was executing at the time of the crash. DB433C40 is the userid of the person signed on at the time. SENECA#RV1 was the name of the workstation that was used at the time of the crash. 659632 is the job number that uniquely identifies the interactive job. We're told when DB433C40 signed on.

The 2nd line of the Black Screen of Death tells us what happened. The program STARTUP was running and crashed with the Error Message CPF2103. Every CL command has it's associated error message. (C D I R) are the options that we have at our disposal. C means Cancel. D means Dump. I means Ignore. R means Retry.

To find out more about the error message, move your cursor to the 2nd line and press F1 for help.

```
                       Additional Message Information

Message ID . . . . . :   CPA0702
Date sent  . . . . . :   09/09/12      Time sent  . . . . . :   09:14:37

Message . . . . :   CPF2103 received by procedure STARTUP. (C D I R)

Cause . . . . . :   ILE Control language (CL) procedure STARTUP in module
  STARTUP in program STARTUP in library DY433D40 detected an error at statement
  number 0000000400.  Message text for CPF2103 is: Library BCI433LIB already
  exists in library list.  Use F10 (if available) or the Display Job Log
  (DSPJOBLOG) command to see the messages in the job log for a more complete
  description of what caused the error.  If you still are unable to solve the
  problem, please contact your technical support person.
Recovery  . . . :   This inquiry message can be avoided by changing the
  procedure. Monitor for the error (MONMSG command) and perform error recovery
  within the procedure.  To continue, choose a reply value.
Possible choices for replying to message . . . . . :
                                                                      More...
Press Enter to continue.

F1=Help    F3=Exit    F6=Print       F9=Display message details
F10=Display messages in job log    F12=Cancel    F21=Select assistance level
```

As we can see from this screen, CPF2103 means that BCI433LIB already exists in your library list. Does this make sense? We called STARTUP twice. The first time added BCI433LIB to our library list. The second time, the library already existed on the list, so the program crashed. Press F10 for more information.

```
                          Display All Messages
                                                              System:   ZEUS
 Job . . :   QPADEV000B    User . . :   DY433D40        Number . . . :   659632

 4>> CALL DY433D40/STARTUP

Print at bottom of each spooled file page: DY433D40 RUSSELL PANGBORN
Library BCI433LIB already exists in library list.
Function check. CPF2103 unmonitored by STARTUP2 at statement 0000000400,
  instruction X'0000'.
CPF2103 received by procedure STARTUP. (C D I R)
CPF2103 received by procedure STARTUP. (C D I R)

                                                                        Bottom
 Press Enter to continue.

 F3=Exit    F5=Refresh    F12=Cancel    F17=Top    F18=Bottom
```

This screen tells us the line number that the program crashed at. The source code for the program used to write the lab looks like this:

```
Columns . . . :    1  71            Edit                        DY433D40/QCLLESRC
 SEU==>                                                           STARTUP
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
       *************** Beginning of data ************************************
0001.00 pgm
0002.00
0003.00    Call StrJob
0004.00    AddLibLE BCI433Lib
0005.00
0006.00 endpgm
```

See where the 400 came from? What line did your program crash at? ADDLIBLE BCI433LIB
Press F12 until you return to the original Black Screen of Death. We have the choice to C – Cancel, D – Dump, I – Ignore and R – Retry.

Typing R in the reply would ask the program to retry the command. Since nothing's changed, there's no point in R.

Typing C in the reply would stop the program from continuing. This is the usual answer to the Black Screen of Death.

Typing D in the reply would stop the program from continuing and give us a report (spooled file) listing the contents of all the variables at the time of the crash.

We'll type I and press enter. I ignores the command and continues executing the program. The next line executed would be the ENDPGM.

Now let's fix the problem.

We can stop Black Screens of Death with the MONitor MeSsaGe command. At the command line type MONMSG and press F4.

```
                          Monitor Message (MONMSG)

Type choices, press Enter.

Message identifier . . . . . .   _________   Name
               + for more values
Comparison data  . . . . . . .   *NONE
Command to execute . . . . . .   __________

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
Command MONMSG not allowed in this setting.
```

The statement '`Command MONMSG not allowed in this setting.`' means that this command does not work in the interactive environment and can only be used in a program.

The monitor message command monitors for specific error messages, traps the error message and executes whatever is in the command to execute. To use this command, you must place it right after the command that you are concerned about. Press F12 to cancel from this screen.

At the command line, type the ADDLIBLE command and press F4. Move your cursor to a black portion of the screen and press F1 for help. This will give you the extended help for the ADDLIBLE entry command. Page down until you see *ESCAPE messages. This is the list of error messages that the ADDLIBLE command can generate – CPF2103 is on the list!

Return to editing your STARTUP program. Use the MONMSG command to stop the black screen of death when the program is run twice. Leave the command to execute blank in this case. The next line of code (ENDPGM) will be executed.

Part D – User Profiles

Changing your User profile

If your user profile contains a program name in the initial program parameter, that program will run immediately upon sign on. Change your user profile to run the program STARTUP in your library.

In 'Green Screen' prompt the CHGPRF command: ===> CHGPRF ***F4***

Look for the Initial Program parameter. The name of the program, and the library it is stored in, have to spelled correctly. Otherwise you will not be able to sign in. Let's try an error first.

Say the program is called ST and the library is DA. **Press enter**.

You are given an informational message that indicates these things are not found. If you ignored the message, there will be a problem in signing in.

Try signing out and signing back in. You should get a screen indicating that the **Job ended abnormally**.

If this happens, you do not have to wait for an instructor to fix your account. You just need to enter a third parameter to override what is stored in your user profile.

At the sign on screen, enter *NONE for the program to be run when you sign on.

Now you should be able to sign in. WRKSPLF and examine the QPJOBLOG spooled file – it contains details of every one of your jobs that ended abnormally. Change your user profile to fix the problem. The program should really be STARTUP and the library name should be your student library, DY433snn. Provide the fix, sign off and then sign in again.

BCI433 Lab 4 CLLE Interactive Program (updated Winter 2020)

Note: If you are still having problems signing in, include some safe information for the last two sign on screen parameters. The Menu parameter could be set to MAIN (In case you accidentally put the program name where the initial menu should go, the system will not find an initial menu with that name.) The library parameter on the sign on screen could be set to your student library or QGPL. These details are also in the cs/~BCI433 FAQ

Check your library list. You should always get BCI433LIB as part of your library list when you sign on. Check it out. Is it there? yes

The user profile describes the user to the system. Your password, initial program, initial menu and current library are all available to you to make changes. Some of that user profile description is not available to you for changing. Your status of being enabled or disabled, User Class and special authorities can only be changed by a Security Administrator. They would run the CHGUSRPRF command to change how you are depicted to the system. Only two user classes usually have the ability to run the CHGUSRPRF command. If a Security Administrator does not have access to the user profile object, then they will fail when using the CHGUSRPRF command. A security officer has the *ALLOBJ special authority and may change a user profile without being granted specific access to that user profile object.

Although you are not allowed to run the CHGUSRPRF command, you can investigate it. In your Client Access session type the following replacing DE433C40 with your user profile name:

CHGUSRPRF DE433C40 (Press F4)

Using the help key (F1) by placing in on the appropriate parameter answer the following:

Status *DISABLED

Can a Status of *DISABLED user sign on? No still run a batch job? Yes

What are the user classes? *SYSOPR *USER

*SECOFR *SECADM

*PGMR

What does the QSECURITY system value have to be => in order to enforce the user class? At QSECURITY level 10 or 20

What would you set Limit Capabilities to in order to allow someone to not be able to change their initial program on the sign on screen and still allow them to type commands at the command line?

Limit Capabilities *YES

Press F10 to look at additional parameters and then page down to see them.

A very important parameter that may overrule what is specified in the user class parameter is called Special Authorities. This can be set to multiple values indicating what a user is allowed to do on the system as long as the QSECURITY setting is greater than or equal to thirty.

Press F4 on this parameter to see the options:

The selections shown as other Values are really what provide capabilities for a user on the system. What are they?

*ALLOBJ *SAVSYS *JOBCTL

*SERVICE *SPLCTL *SECADM

*AUDIT *IOSYSCFG

a *SECADM special authority allows someone to change a user profile. What else can they do?

Create, change, and delete user profiles.

What is this special authority and what user class automatically gets this special authority?

The user is given the authority to change, display, hold, release, cancel, and clear all jobs that are running on the system or that are on a job queue or output queue that has OPRCTL (*YES)

Job control (*JOBCTL)

Part E – Develop Display file – LAB4DF

A program is available for you to run. Your task is to duplicate it's features. It is found in BCI433LIB. The safest way to run it is to change your current library, run the sample program and change your current library back.

```
CHGCURLIB  BCI433LIB
CALL LAB4CL20
CHGCURLIB (whatever your usual current library name is)
```

Try the following to experience an unsuccessful run of an earlier lab 4 program version.:

==>CALL LAB4CL11

The program will not run because it is found in the LAB4LIB library.

Try to run the existing program LAB4CL11 in the LAB4LIB library using the qualified name. What did you type?

==> CALL LAB4LIB/LAB4CL11

You should get the following error message:

```
                Display Program Messages

Job 440884/WS540A40/QPADEV000J started on 01/20/11 at 15:28:58 in subsystem
CPF4101 received by procedure LAB4CL11. (C D I R)
```

Put your cursor on the error message and press F1. There is a long explanation, but we can get a quicker one by pressing **F10**

The first line below **CALL LAB4LIB/LAB4CL11** should tell you what the problem is.

What is the problem? Application error. CPF4101 unmonitored by LAB4CL11 at statement

That occurs because the compiled display file that this program uses (LAB4DSP11) is not in any library that is part of your library list.

What is the command that will include LAB4LIB on your library list so when the LAB4CL11 program is running and tries to use the display file, it will be found?

ADDLIBLE BCI433LIB
You know how to successfully run LAB4CL20.

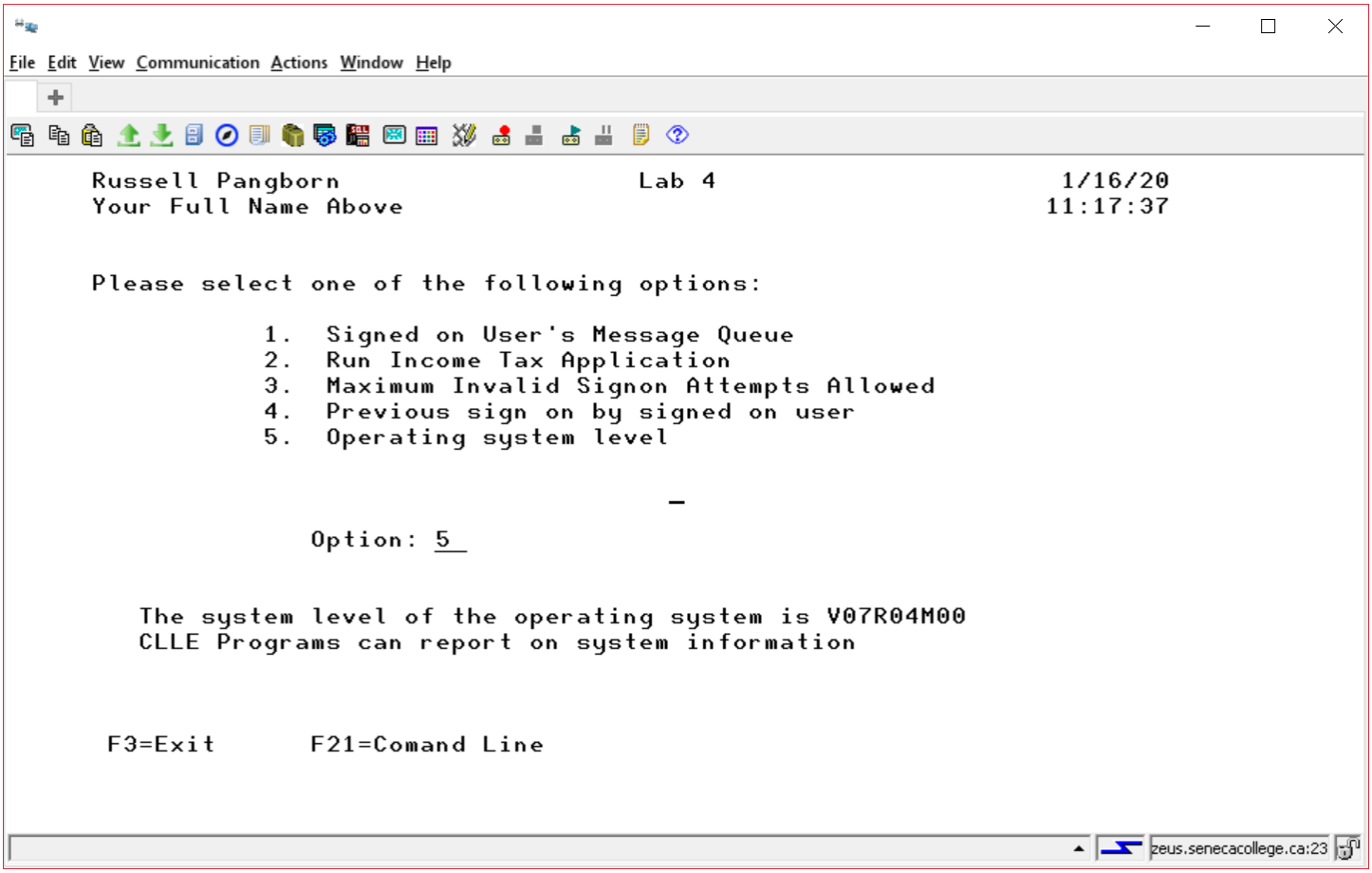
Investigate what this program does**.**

BCI433 Lab 4 CLLE Interactive Program (updated Winter 2020)

Your first task will be to develop the display file. The text has already been done for you. It can be copied from BCI433LIB/QDDSSRC. The member name is LAB4DSPXX. You can change this to LAB4DSP.

Change the text at the top so your name shows in place of “Russell Pangborn”

Include the appropriate fields and constants and enable the appropriate function keys.



Design your display file to include the following:

Function Keys:

F3 – Exit F21 – Command Line F21 must be defined as a CF function key.

Include on the screen:

Your Full Name System Date
System Time Appropriate Title

An input/output field that allows 2 characters. This field is designed to hold the option entered. Possible values are: 1, 2, 3, 4, 5, 01, 02, 03, 04, 05. Include an error message if something else is entered. If there is an error, the option field should be reverse image and the cursor positioned on it.

Two 70 character output only fields. These fields are designed to hold text of for the selected request. The fields can be named MsgTxt1 and MsgTxt2

The OPTION field will use an indicator to show the field in Reverse Image and to position the cursor.

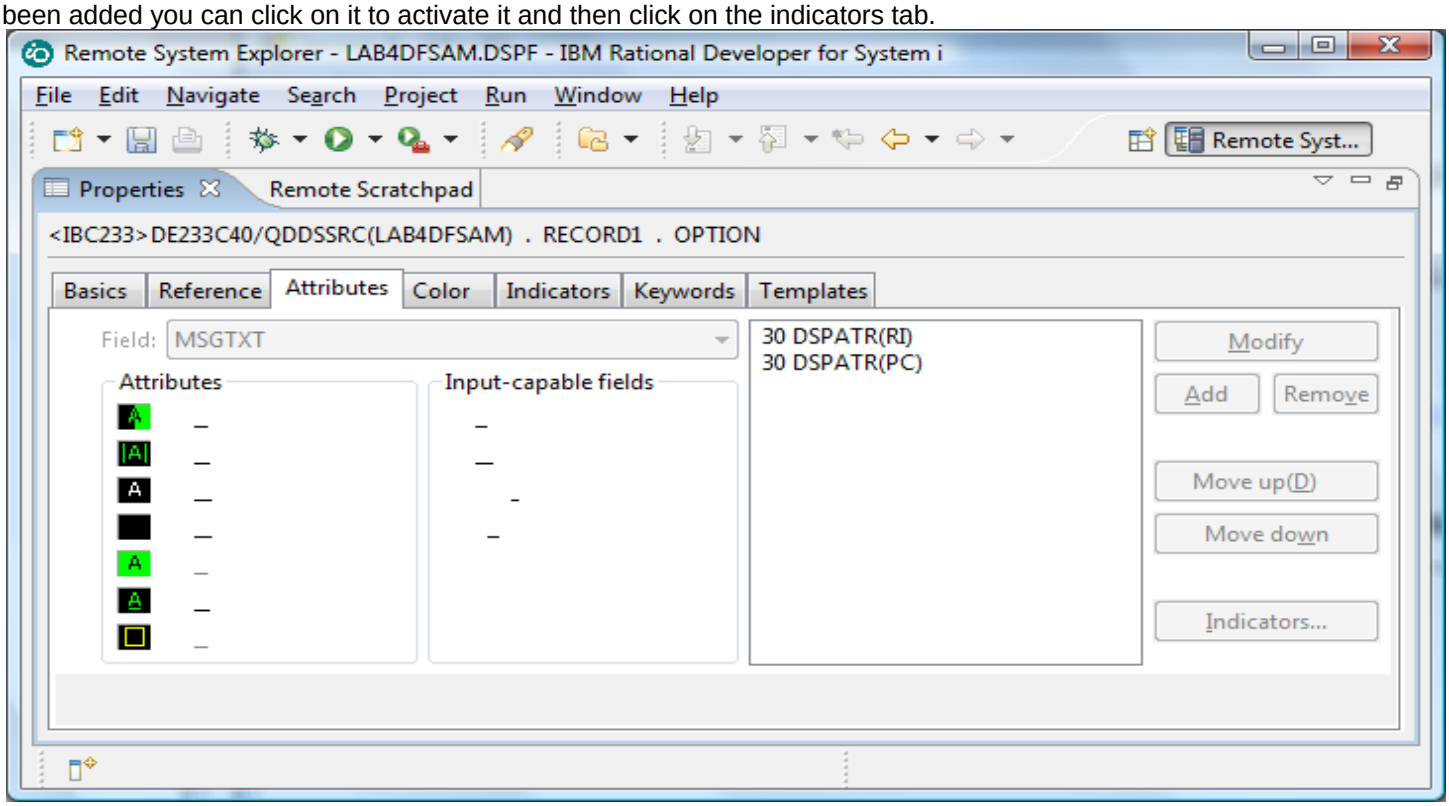
OPTION 2A B 14 25

```
A  30                                  DSPATR(RI)
A  30                                  DSPATR(PC)
```

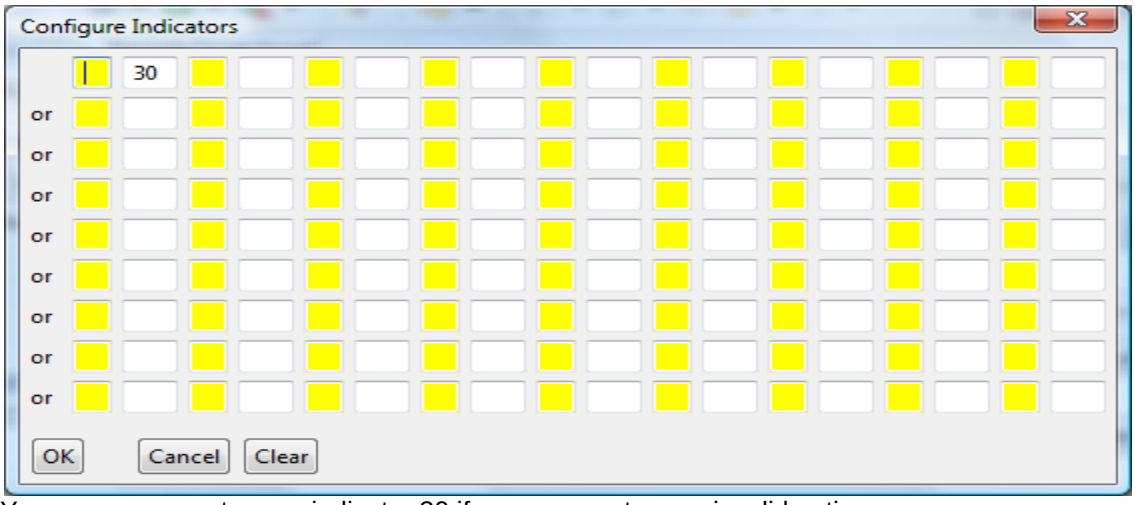
This can be entered carefully below the option field when on the Source tab. Use an "I" to insert a line.

This can also be achieved by using the Properties tab when the focus is on the OPTION field.

The attributes tab is selected and Reverse Image is checked and then an add button is used. Once DSPATR(RI) has been added you can click on it to activate it and then click on the indicators tab.



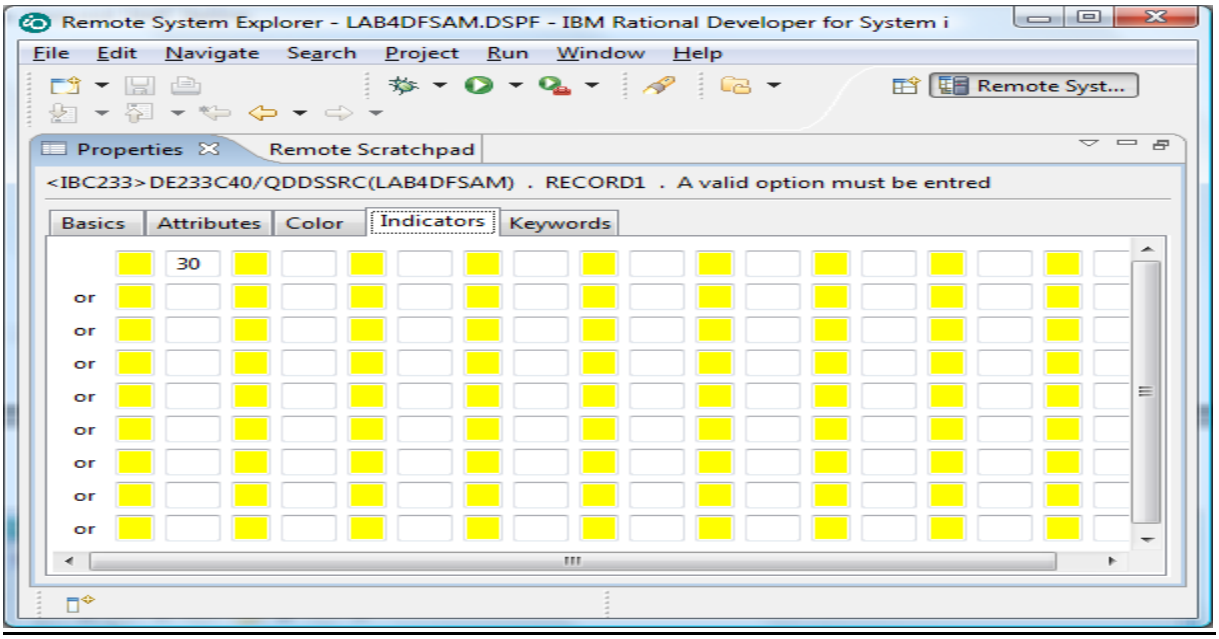
By entering 30 you are saying indicator 30 must be on in order to activate Reverse Image.
If you put an "N" in the first box, you would be saying indicator 30 has to be off in order to activate Reverse Image.



Your program can turn on indicator 30 if someone enters an invalid option.

You can also click on the Text "A valid option must be entered" in order to get the text to appear beside

the option field only when indicator 30 is on.



Part G – Develop CLLE Program – LAB4CL

Here is some skeleton code you can enter that gets option one working from last semester's CLLE lab4.
This code reports on the security level setting.

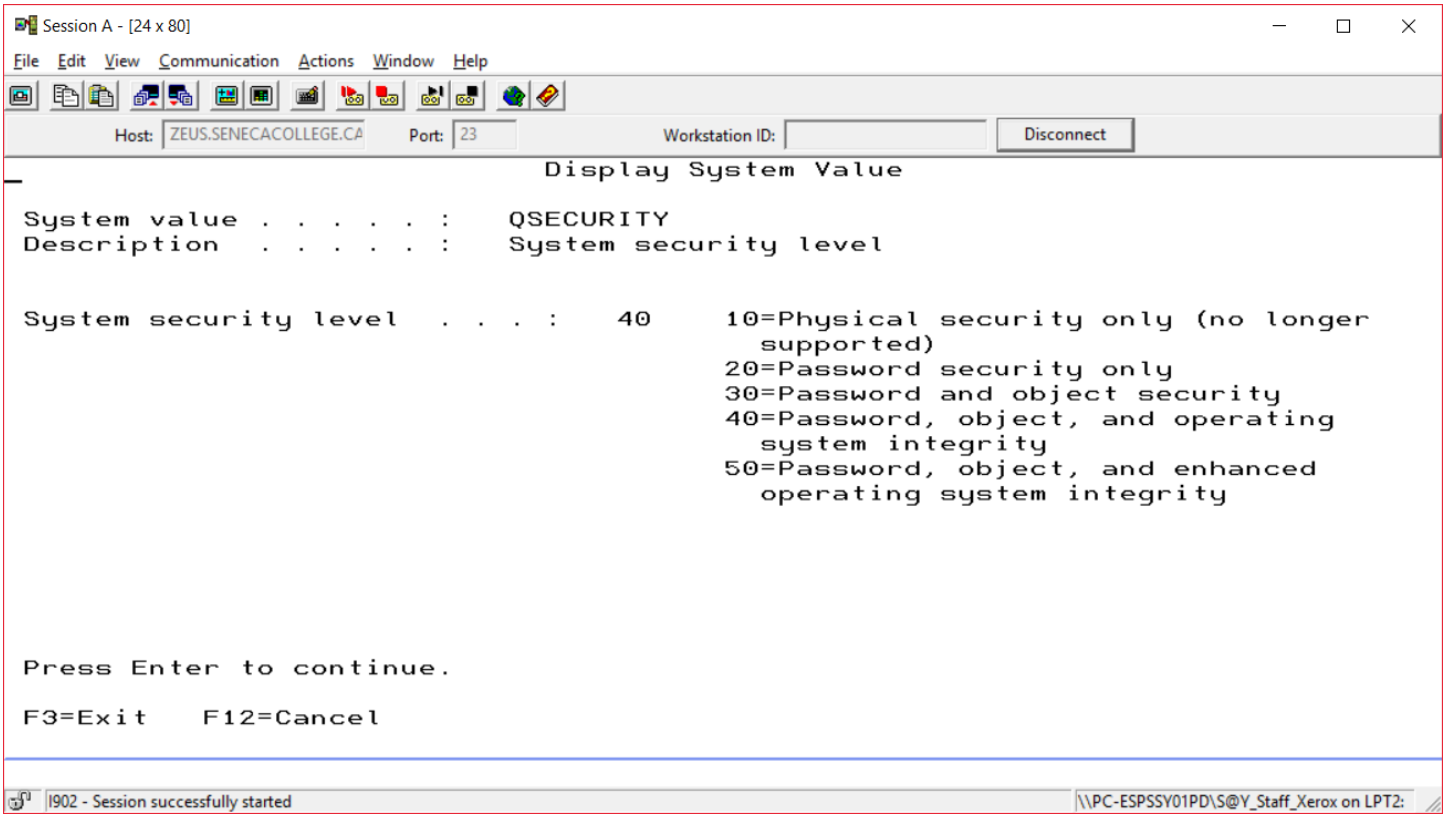
This code has been discussed in class. **It is not the code you will be using in your finished program**. You can just view it, or code it when testing the ideas – but do not show this option when submitting your program.

You need to change it so that it reports information on the security level.

```
PGM
DCLF     LABDF
DCL      &Security    *Char 2
                          /*** M A I N L I N E  *****/
SndRcvf
   DoWhile (&In03 = '0')

      Select
         When (&Option = '1' *or &option = '01') CallSubr OPTION1

         Otherwise ChgVar &in30 '1'
      EndSelect

      SndRcvf
   Enddo
                                  /*** S U B R O U T I N E S  *****/
SUBR OPTION1
```

```
    RtvSysVal SYSVAL(QSecurity) RTNVAR(&Security)
    Select
      When (&Security = '10' ) +
            ChgVar  &MsgTxt  ('Security Level is 10  and not supported')
      When (&Security = '20' )  +
            ChgVar  &MsgTxt  ('Password security only at level' *Bcat  &Security)
      When (&Security = '30' ) +
           ChgVar    &MsgTxt ('Password and object security at level 30')
      When (&Security = '40' ) +
           ChgVar    &MsgTxt ('Password, object, and operating system integrity at level 40')
      When (&Security = '50' ) +
           ChgVar    &MsgTxt  ('Password, object, and enhanced operating +
                       system integrity at level 50' )
   EndSelect
  EndSubr
ENDPGM
```

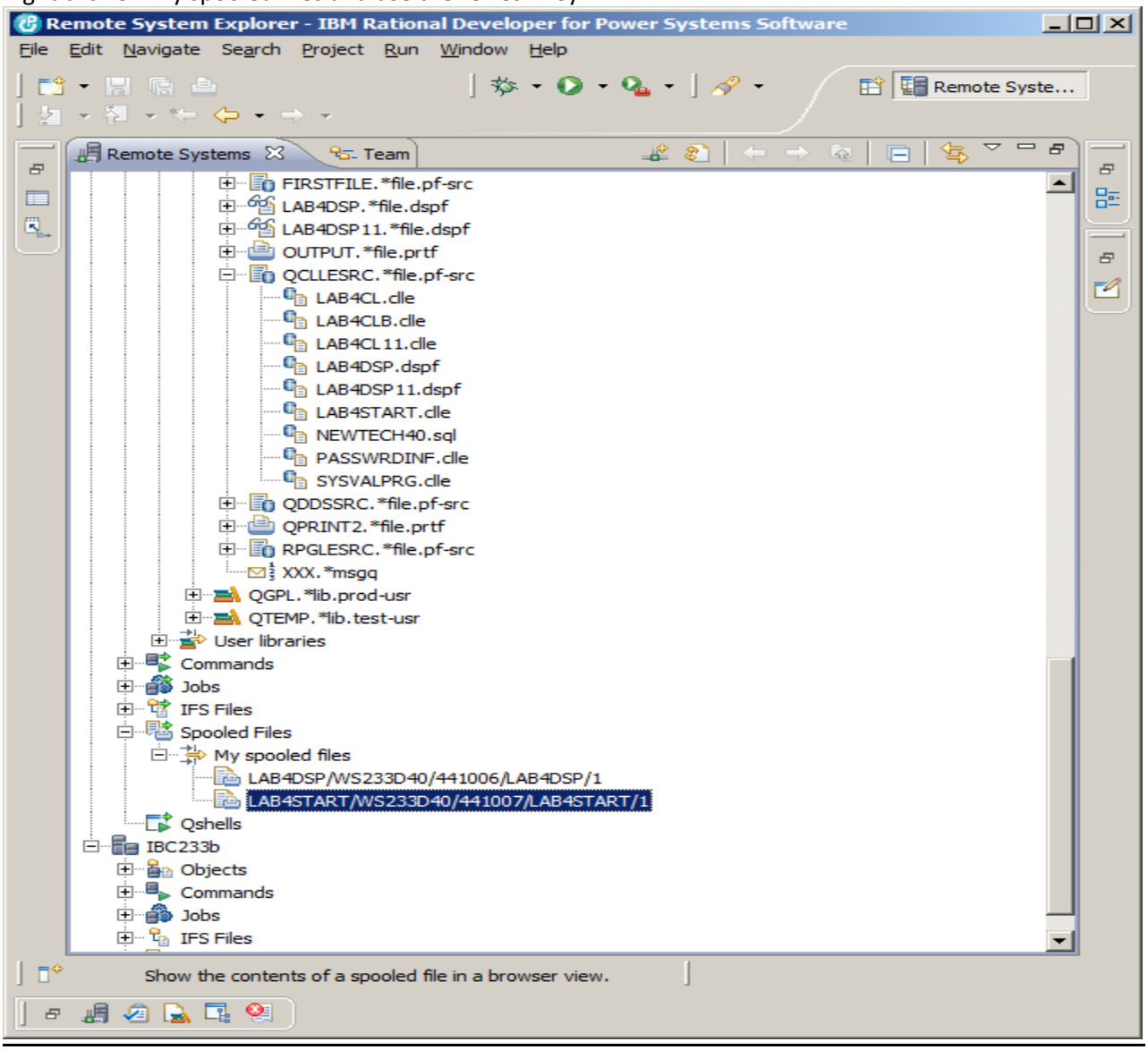
The above code was figured out by running DSPSYSVAL QSECURITY and taking the screen information to be applied based on the number that is returned by this command and inserting it into a message text field.



Important info: The above example used the RTVSYSVAL command to load a CLLE variable with information. Other RTV commands that may be useful are RTVJOBA, RTVUSRPRF, RTVOBJD, and RTVNETA. In order to find the current operating system level, you need to look at QSYS/QCMD with the RTVOBJD command.

Let's look at the spooled files generated when we compile in RDi

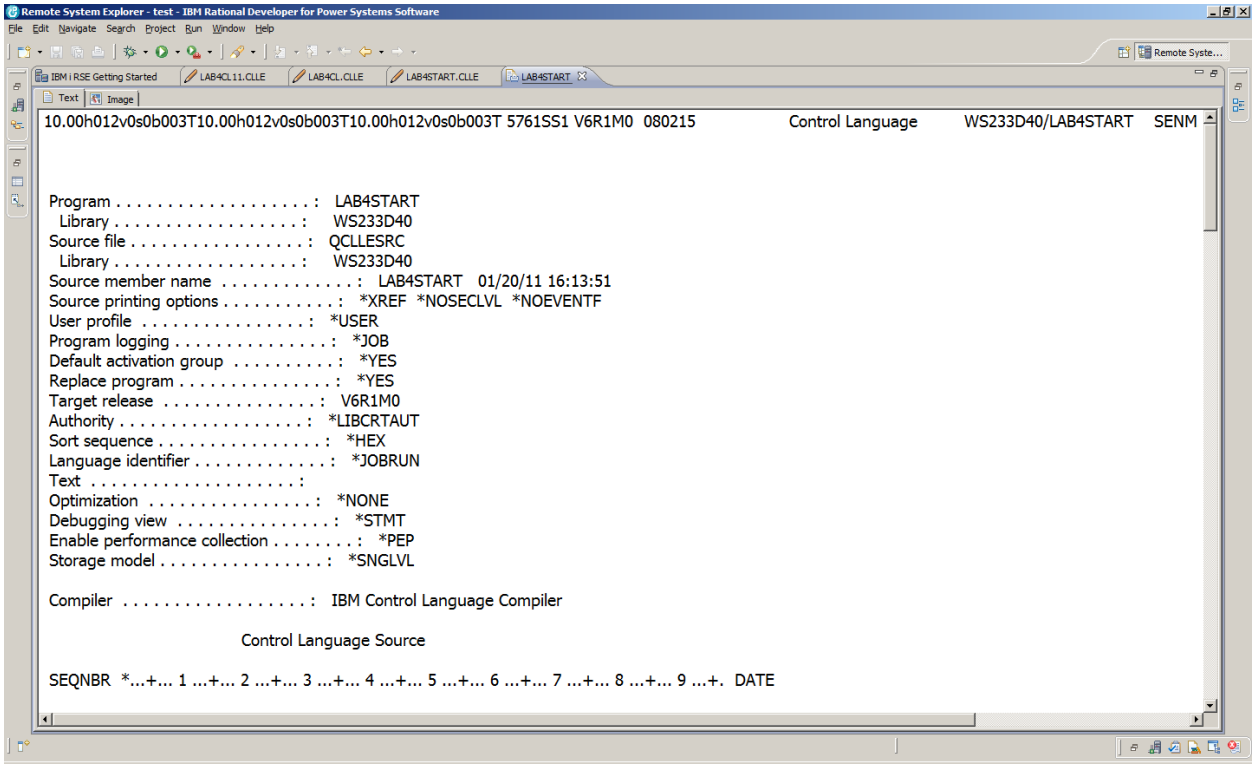
Right click on My spooled files and use the refresh key



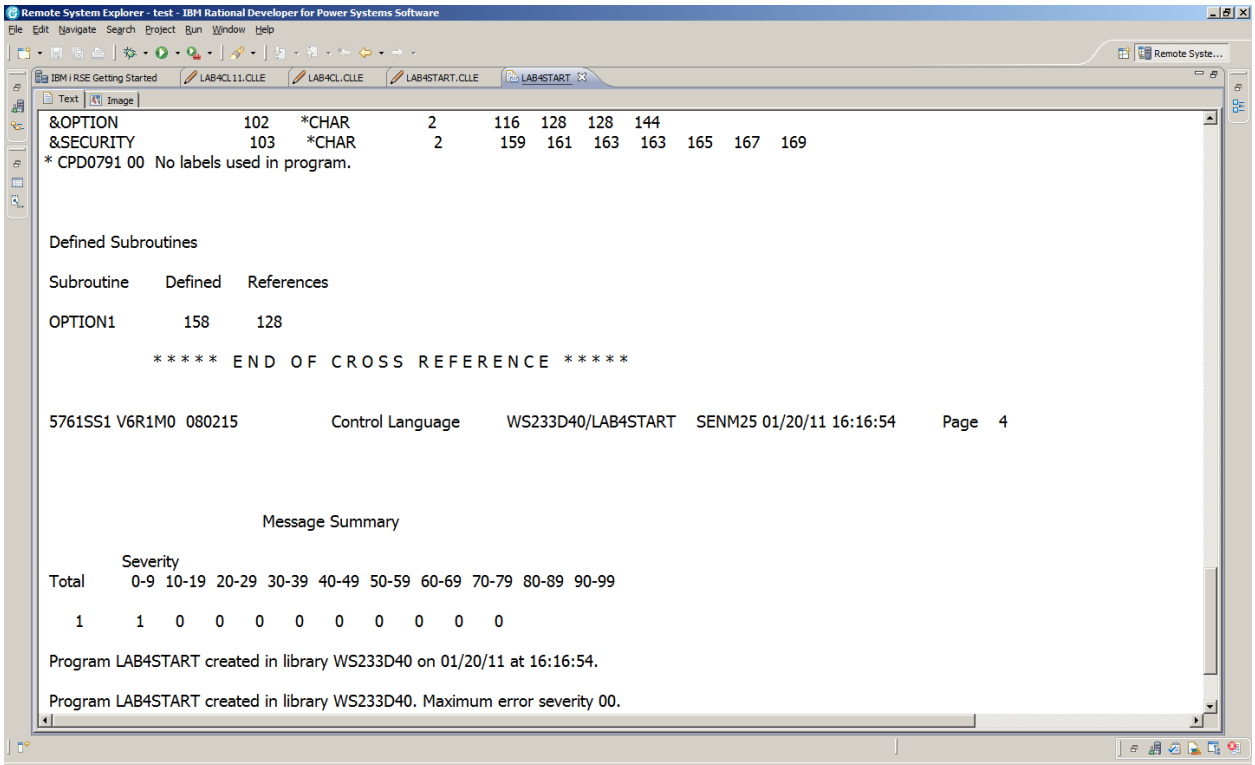
Select your spooled file from your CLLE compile and right click to select "Show the spooled file contents"

My spooled file and program were called LAB4START. I can double click on the tab to see the Text of my listing.

(The other option is an Image tab beside the Text tab – but you may have to wait a bit for this)



Text is good enough for me and I can scroll down to see if the program object was created in my library.

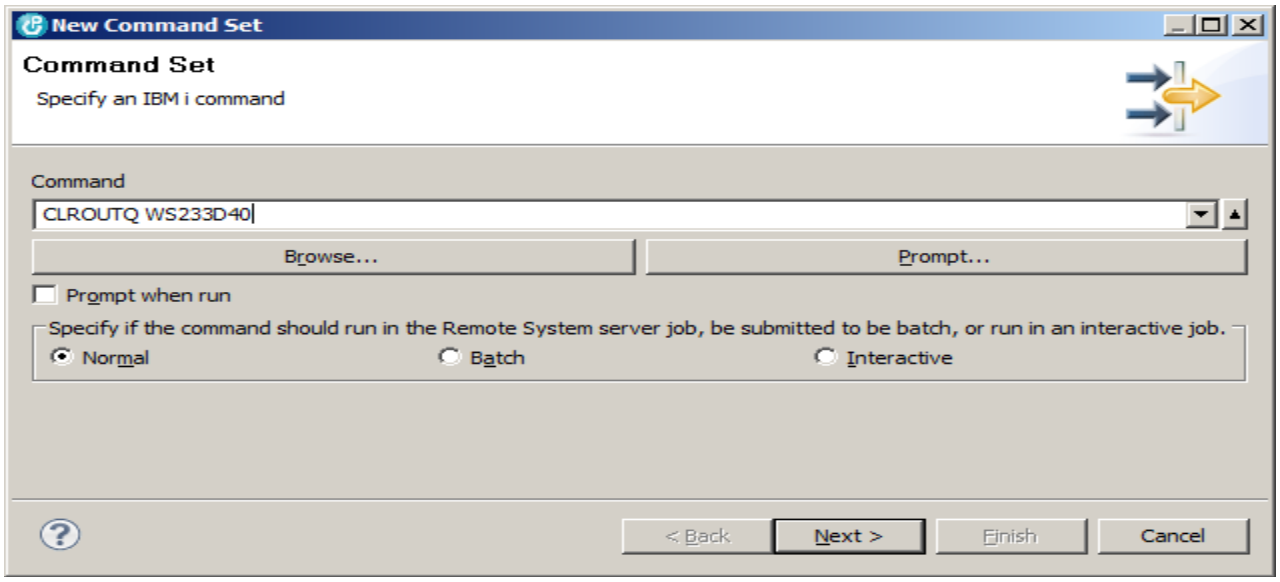


Create a Command to get rid of all spooled files.

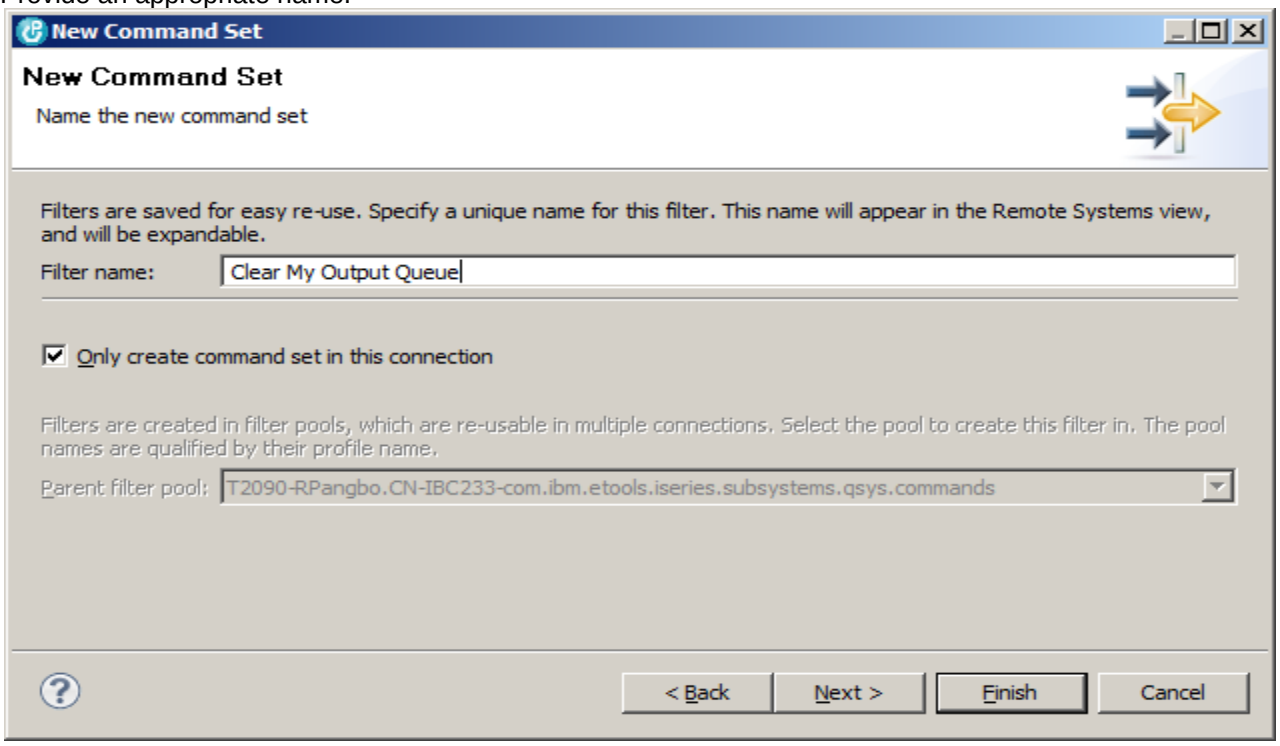
Unfortunately as we work things out, there will be a lot of unsuccessful compiles. The output queue gets crowded and it is tough to find the latest spooled file. We can get rid of most of them in RDi with the CLROUTQ command. (Clear OutQ)

This command is handy to have always as part of our saved workspace.

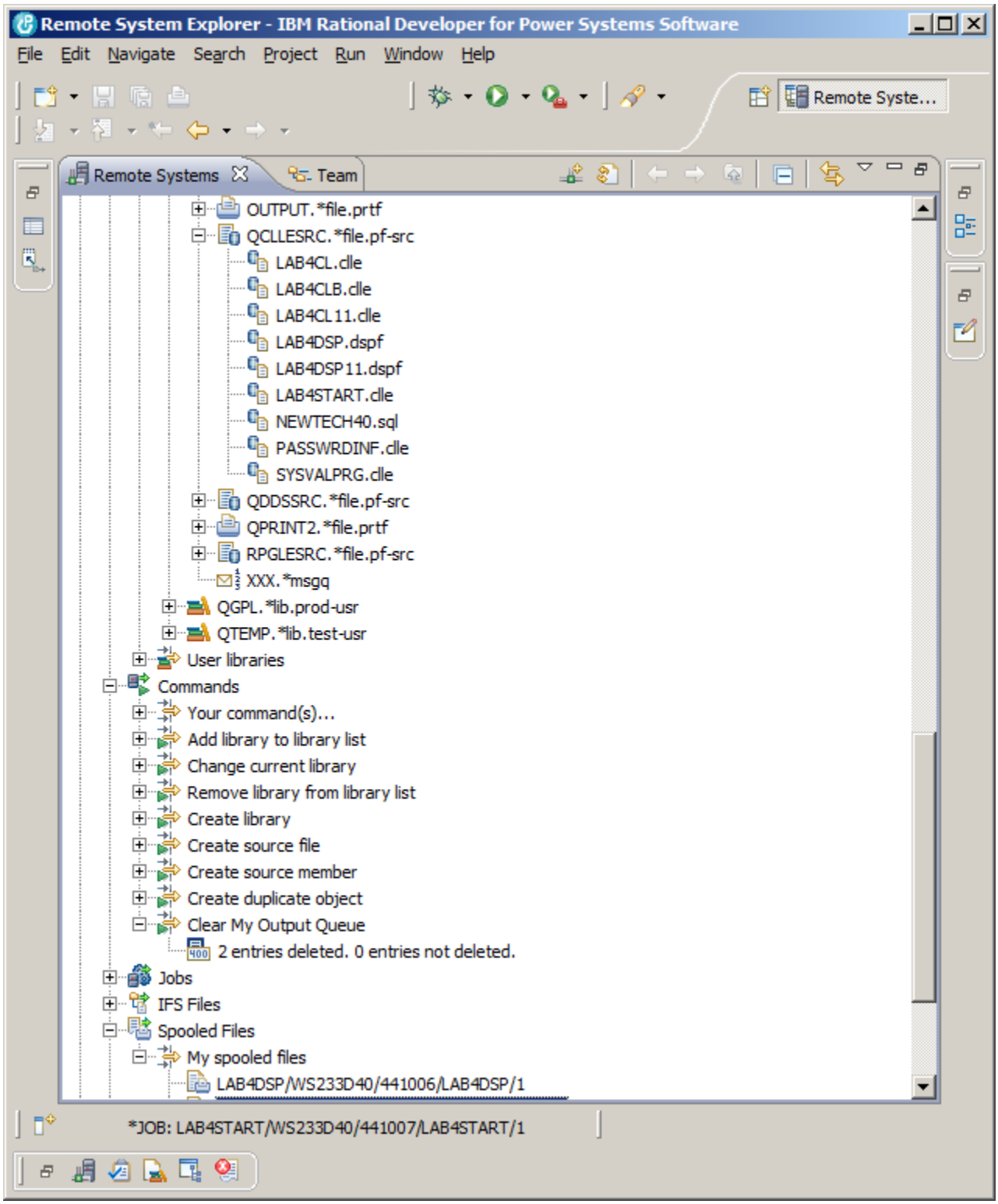
In the Remote Systems tab, Expand **Your Commands** and provide the command to clear out your output queue.



Provide an appropriate name.



Run your command.



If you have your workspace on a USB or on a laptop, it should be available next session to deal with your cluttered output queue. Don't collect a lot of old spooled files in your output queue!