

BCI433 – Lab 2

Summer 2018

OBJECTIVE:

- Introduction to RDi (IBM Rational Developer for System i) and Eclipse
- **Create and successfully compile an RPGLE program**
- Print out a compiled listing
- **Create a physical file called STUDENTS using DDS**
- Enter Student Data using into your physical file using DFU
- **Display the Student Data with RDi, SQL and the RUNQRY command**

Requirements to pass the lab:

Save your workspace on a USB drive / flash/thumb/jump drive / memory key
Demonstrate a Lab configured Workspace
Demonstrate how to get a library always added to your connection's library list
Demonstrate how to get a command run when signing on to an RDi connection
Demonstrate using RDi a successfully created STUDENTS file with your name in one of the records
Answer questions on this handout

Note: We use the term 'Green Screen' to refer to which ever emulator tool you are using. (Client Access)

Make sure that your machine was restarted before you use it!!

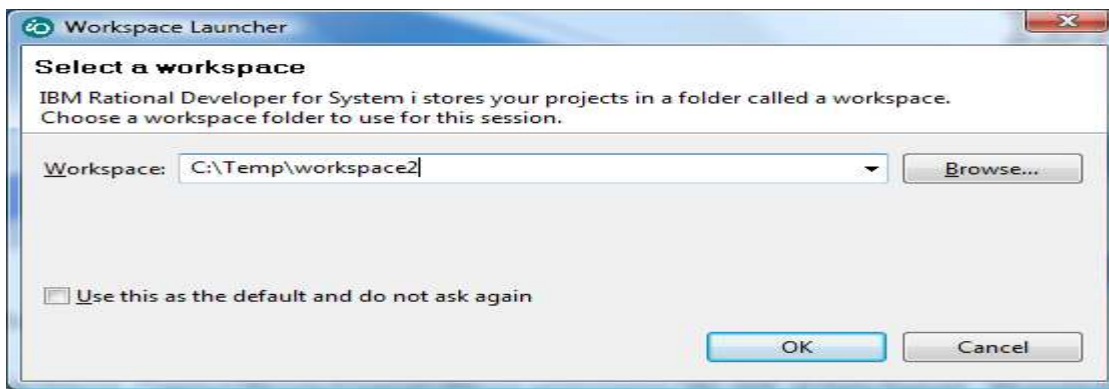
We'll be using RDi in this lab. RDi stores the development environment on your PC. The default workspace is a folder in the temp directory.

Unfortunately, the default workspace that's been provided for you has errors, so we'll be creating a workspace from scratch. It's very important that you protect your workspace and keep it on your USB drive. SHUT YOUR LAB WORKSTATION DOWN when you've finished working!

Once you are done, copy your workspace to your USB drive so that you'll have all of your settings available when you're ready to start working again.

1. **Start IBM Rational Developer for System i.** Your instructor has demonstrated how this product can be started. Click on the Start menu. Click on All Programs and then on IBM Software Development Platform. You should be able to navigate the rest of the way. Ask your instructor if there is a shortcut folder available to start this product.

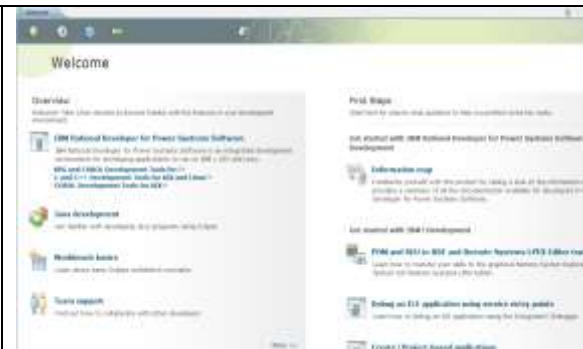
Depending upon how your computer is configured, you may be prompted for your default workspace:



e.g. a folder on desktop on Windows

Change the path to something unique to you. It's a good idea (for speed) to temporarily use the temp folder and then copy the workspace folder to your USB drive when you're done.

If this is the first time opening the software on this computer, you will see the welcome menu:

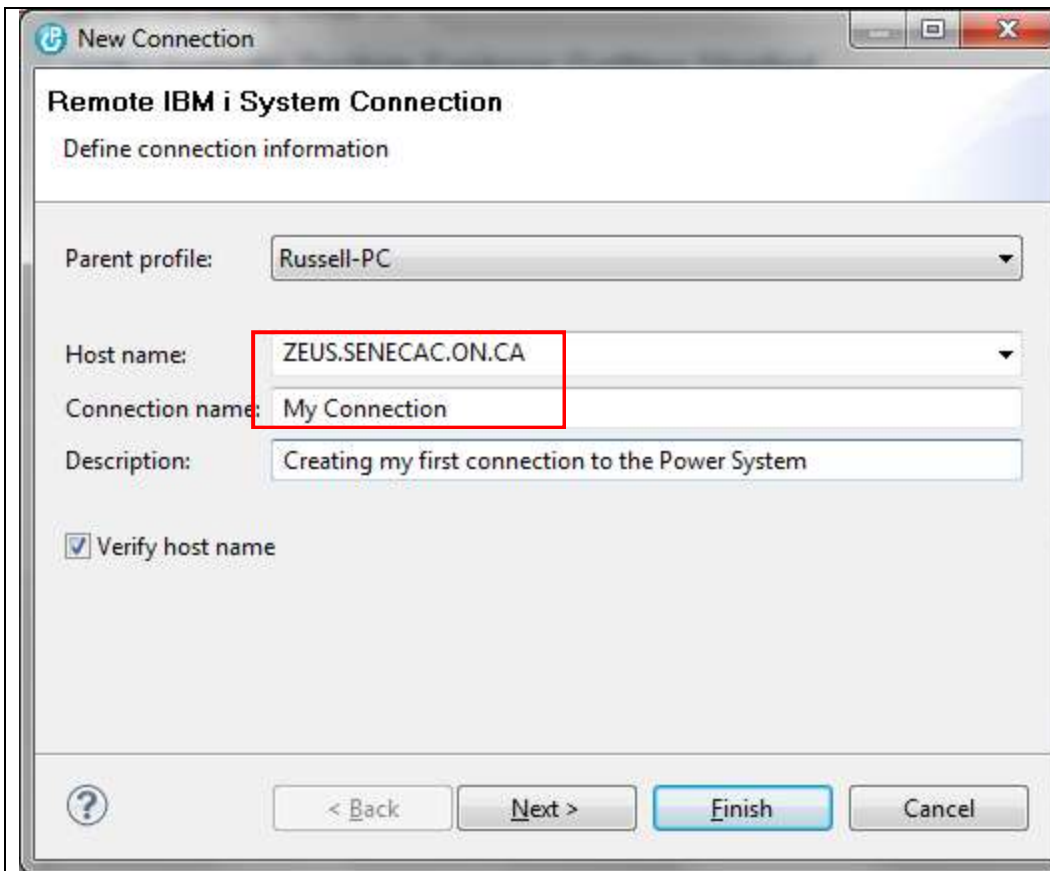


Close this window by clicking on the X

Don't forget about this as a reference. You can investigate some of the things discussed in the labs and the textbook. You can also find hidden gems that are useful that haven't been covered by your labs.

2. You should see the following in the remote systems explorer view.





Expand New Connection and then **IBM i**.

You can call this connection MyConnection beside Connection name.

The Host name is:

ZEUS.SENECAC.ON.CA

When you set up RDi at home you will need to do this. You may opt for a more meaningful connection name at that time. (maybe **BCI433**)

Click on Finish when you are done.

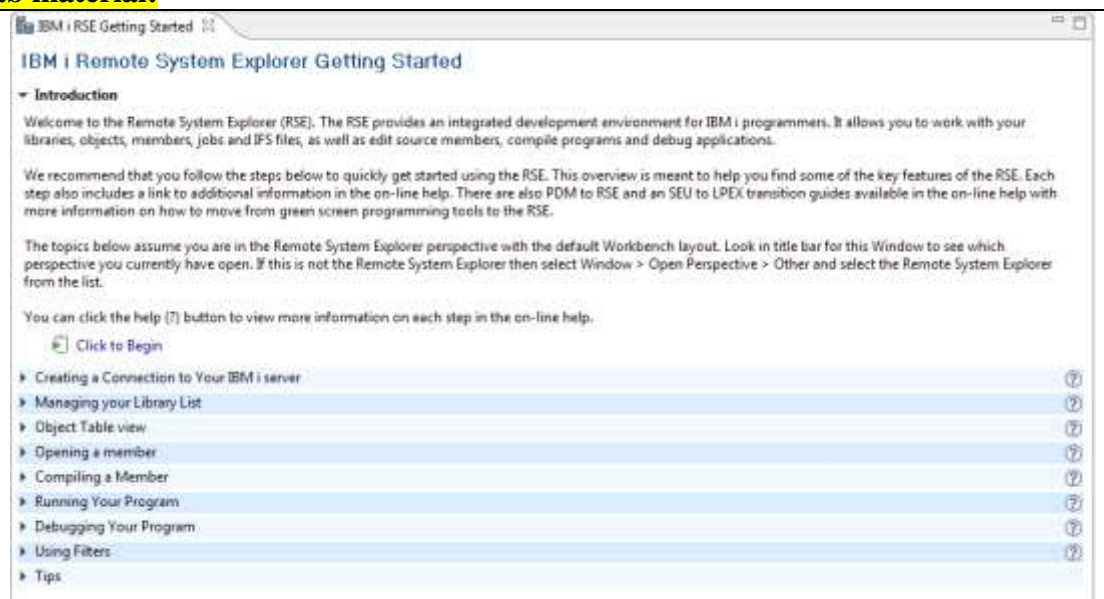
3. **Help outside of lab material!**

Click on Help in the top menu, then click on IBM i RSE Getting Started.

This should give a full screen view of help provided for Remote System Explorer.

Click on Creating a Connection to Your IBM i server. We just covered that – but maybe you prefer this description.

Use this as an additional resource to the labs. It is always available to you.



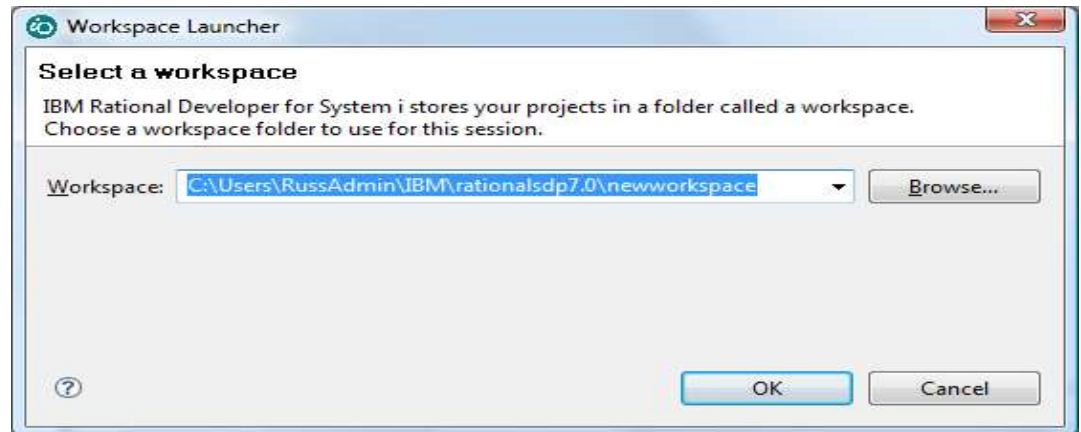
The Workspace.

When working with RDi, your configuration and temporary versions of programming source code **is stored in a "workspace"**. It's just a folder with a lot of files in it. Since students at Seneca do not always use the same machines, the workspace is deleted when you reboot a machine. Full time developers do not have to contend with this problem.

If you want to use a different workspace than the current one, you can switch workspaces.

Click on File and then **switch workspace.**

A popup window like this one should be displayed:

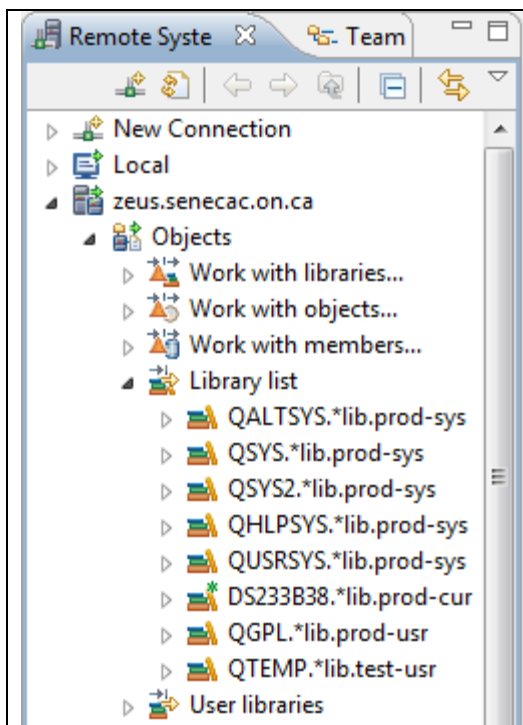


If you find the I/O to the USB drive is slow, copy your USB workspace to the local drive before answering the above dialog, e.g. C:\TEMP2\MYWORKSPACE, then specify that location above. **When the RDi session is finished**, copy the workspace back to your USB drive – don't forget otherwise you will have to do it all over again.

USB drives are quite robust but they have been known to fail. Always back up your data. A backup is a copy in a geographically separate location. Having two USB drives is fine but not when both of them are in the same backpack. Copy the contents of your USB drive to your home computer – and don't leave the USB drive at home. An alternative is to zip the USB drive data to a USBbackup.zip file and email the zip file to yourself or ftp it to a server somewhere.

You could use the default configuration each time you start up the session. Although any customization you do will be lost, if you restrict yourself to working with just the program code stored on the server, i.e. Zeus, then your work will be preserved. People use this method when they forget their USB drive. Remember, you will be starting from a basic default configuration.

Sign on to RDi (IBM Rational Developer for Power Systems by expanding the connection that you just created and then expanding objects and then finally expanding the library list).



When you expand your library list you will be asked to sign on.

The first time this dialogue shows it is a good idea to check off Save User ID.

It is not a good idea to check off Save password



What libraries does your library list contain for the connection you opened?

QALTSYS.*lib.prod-sys; QSYS.*lib.prod-sys; QSYS2.*lib.prod-sys; QHLPSYS.*lib.prod-sys;
DC233C02.*lib.prod-cur; QGPL.*lib.prod-usr; ~~QGPL.*lib.prod-usr~~; QTEMP.*lib.test-usr

Sign on to the iSeries using the Client Access emulation tool and determine **the objects you have in your library**. ==> **WRKOBJPDM DS233smn** (substitute your library name here)

Object	Type	Attribute
SYSVALPRG	*PGM	CLLE
DC233C02	*OUTQ	
QCLLESRC	*FILE	PF-SRC

There are a few objects on the system with the same name as your user id. Find them with this command:
==> **WRKOBJ** use **F4** to prompt it, **enter your User ID as the "Object"**. **Accept the defaults and press enter.**
SM433A36

For the DS233smn object there are **the following objects with the same name:**

Object type are:

*LIB
*USRPRF
*MSGQ
*OUTQ

Libraries containing objects:

QSYS
QSYS
QUSRSYS
DC233C02

To investigate the WRKOBJ command type it again at the command line and press F4

Press **F1** (the help key) and **F2** (extended help)

The first sentence should tell you what the WRKOBJ command does.

The Work with Objects (WRKOBJ) display shows a list of _____ you

have (*USE) authority to use.

We will get into the habit of **switching between** 'Green Screen (Client Access)' and IBM Rational Developer for System i (RDi) which shows in the taskbar as "Remote System Explorer". You have just completed a task in 'Green Screen' and are ready to go to the RDi screen. Use ALT+TAB to find RDi's "Remote System Explorer" or click on the taskbar icon.

Lets go back to the RDi session

Remember - just like the green screen, RDi has a lot of help built in to it. What were the two help features referred to and seen earlier in this lab that were available from the Help Menu?

Menu bar > Help > Welcom

Menu bar > Help > Getting Started

You will find a lot of context sensitive help available **in RDi also**.

Five views are included for you. The tab in the upper left is the Remote Systems view. This view shows the connection that you just built.

There is a Properties view at the bottom left, a view for showing detailed information at the bottom right, an empty view over top (this will be used for showing your code) and an Outline view in the top right.

All of these views together comprise **a Perspective**. Perspectives can be changed to include other views or to show the views in a different order (for example the Outline view could be moved to the bottom left of the screen). There are some predefined perspectives (for example debug) and you can customize your own perspective.

4. Library Lists.

Let's look at some of the things that you did in Lab 1 and see how it's handled in this environment. You have displayed the library list for the BCI433 connection.

What is your current library?

e.g. [SM433A36](#)

What command did you have to use in Lab1 to show the same information?

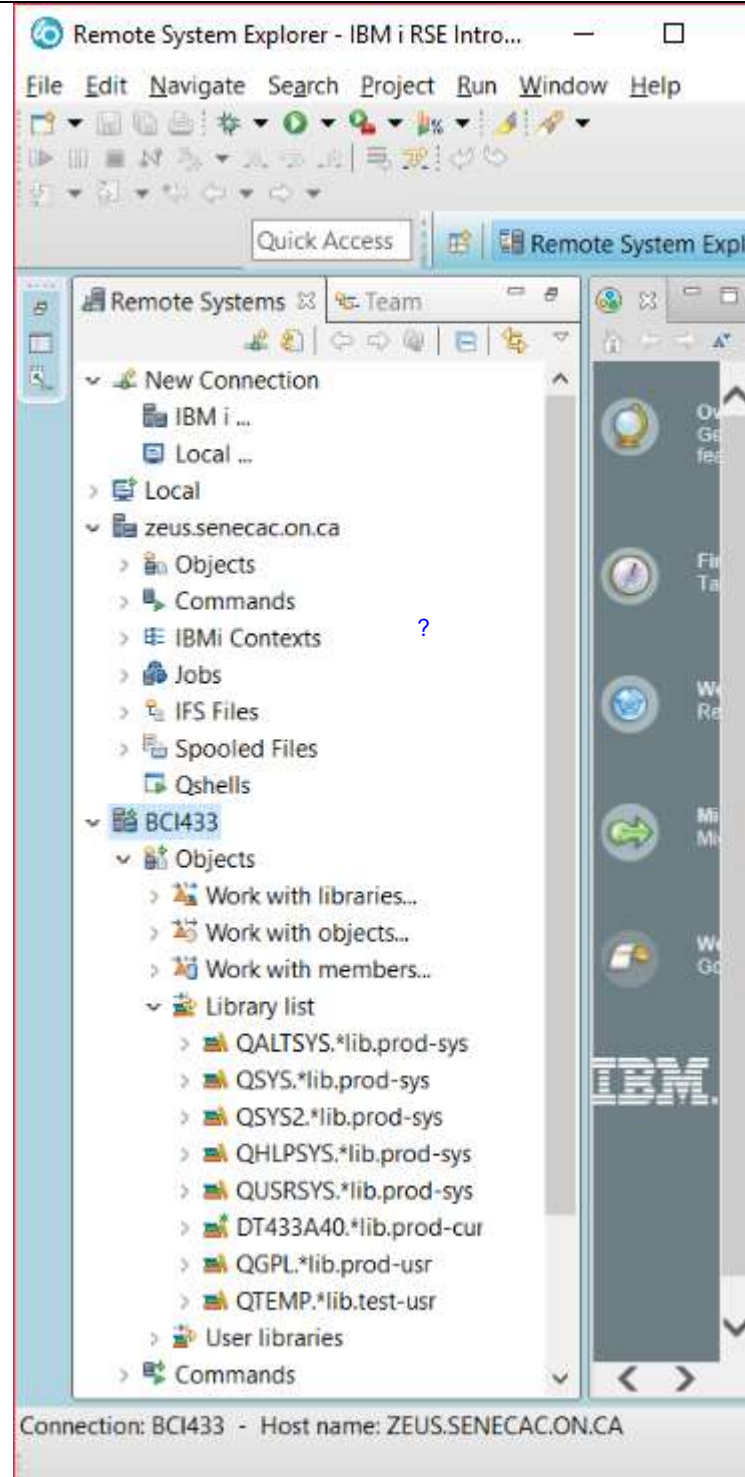
?

Click on the plus sign beside your current library. What objects do you see?

[SYSVALPRG.*pgm.clle](#) [SM433A36.*outq](#)

[QCLLESRC.*file.pf-src](#)

Click on the plus sign beside the QCLLESRC file you created in the first lab. What is the difference between [SYSVALPRG.clle](#) and [SYSVALPRG.*PGM.clle](#)?



Double click on [SYSVALPRG.clle](#). You should see the code from your first program in a view pane. There is not much code to maintain for this program. You can enter a new line and still see all the existing code. If this was a large program, you may wish to have more screen space to work with.

Double click on [the SYSVALPRG.CCLE tab](#) in the editor. Now you should see just your program and not the other views. Double click again on this tab. Now you should see your program and the other views.

Initial Library Lists

You can create multiple connections for a single server in one workspace. This is handy when you are taking multiple courses that use Zeus! Each connection can have a different library list used to find unqualified object names. You can sign on several times onto Client Access sessions and also have a different library lists in each session by adding or removing libraries from the list with CL commands. Library lists are built when you sign on and destroyed when you signoff, so remember, if you change your library list in RDi, you should change it in Client Access for consistency sake!

For all sessions there is a default set of libraries on the list when you sign on. This default list can be changed.

RDi is for software development. If I am compiling a program stored in my library that needs to access an object in BCI433LIB during the compile process, I will get an error message because the object required by the compile is not in my library or any of the libraries in the connection library list.

Let's try this. We will code a simple program that requires a file called **STUDACCTS**. STUDACCTS is found in the BCI433LIB library.

Here is our silly little program that refers to an object that is not in your library or any of the other libraries on the library list for your connection. This program will not do anything observable. It is just being used to illustrate how library list settings can affect the success of a compile. It is an RPGLE program. This is the code:

```
ESTUDACCTS IP E      DISK
```

With RPGLE programs this line that describes the data file needs to have entries made in specific columns. **E** goes in column 6, **I** goes in column 15 etc.

You are not expected to count column positions out. Instead a prompter is available for you – keep reading – we'll show you how to use it!

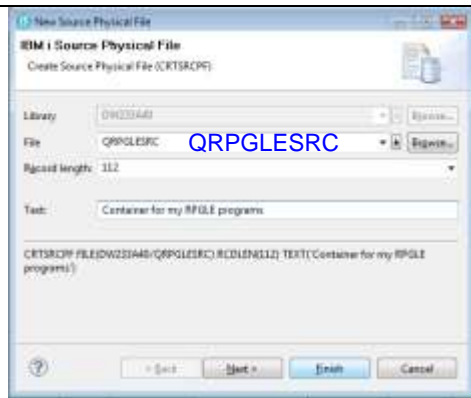
We will first enter the code and try compiling it when BCI433LIB is not on your connection library list. Then, we will add BCI433LIB to the library list and recompile.

After a successful compile, we will close up RDi and restart it and immediately retry the compile.

5. Entering Code with RDi

You can work with several programming languages from the same environment. You could do your OOP244 course work using RDi on the iSeries. You would just need to create a source physical file for C code. RPG, CLLE, COBOL and JAVA are used frequently on the System i. We will create a second source physical file for our lab 2 program. This container will be use to hold RPGLE programs.

Do this by right clicking on your current library. Select New and then select Source Physical File.



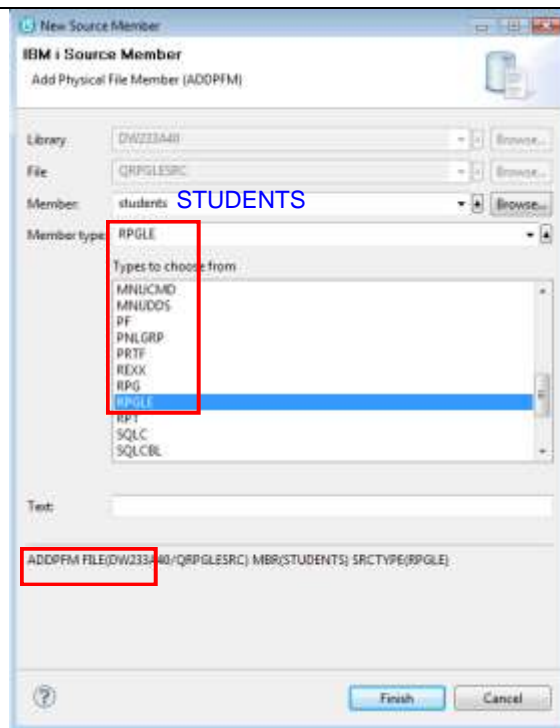
What was the keyword notation for the CL command you entered primarily by pointing and clicking? (Look at the bottom of the iSeries Source Physical File box just below Text)

CRTSRCPF FILE(SM433A36/QRPGLSRC)

Enter the file name and text comment and then select Finish.

Take a look at the Commands Log. You should see the command you ran and some feedback indicating if it ran successfully.

Right click on QRPGLSRC.*file.pf-src and add a new member called **STUDENTS**.

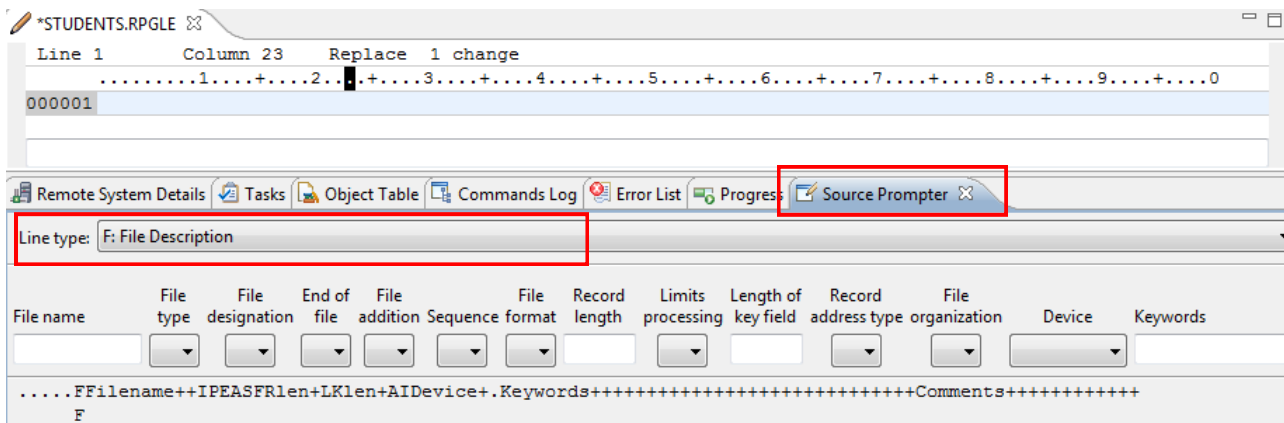


<-----ADDPFM

This is the keyword notation for the command to add a physical file member.

When the ADDPFM command is finished executing you should see that the LPEX editor is ready for your code.

A. **Press F4** in the area where you need to enter code. Then **double click on the Source Prompter tab that appears**. You will use the drop down arrow where it says **Line Type** and **select F File Description** as you prompt type. (this screen shot shows the Source Prompter tab before double clicking on it)

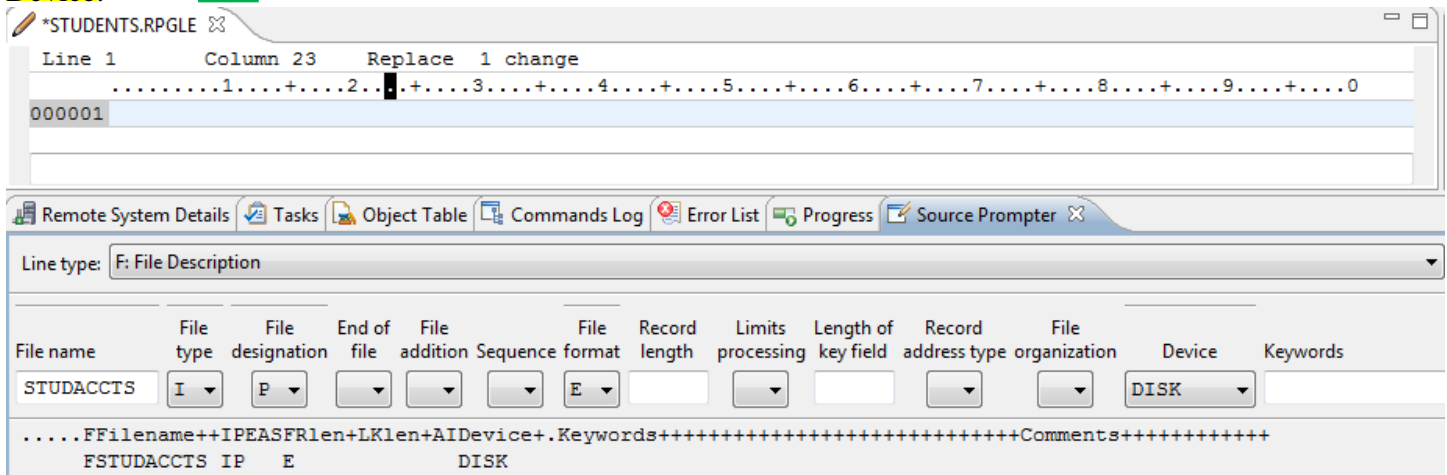


A screen shot of how to enter this code has already been below. You will be entering the following:

File name: STUDACCTS **File Type:** I (means input)

File Designation: P (??) **File Format:** E (means externally described)

Device: Disk



Apply

After you have entered your code, click on the **Apply** Tab.

You can toggle between just showing the source prompter view and a perspective showing the source prompter, the entered code, the Remote Systems Explorer views etc. It is nice to have easy access for up to 12 tabs with the overall perspective. Sometimes you require more room on the screen when working with a view. All you need to do is **double click on the tab** in question to **toggle between a single large view or several smaller views on your screen**. Try this toggle feature out with the STUDENTS.RPGLE , Remote Systems, Commands log and Properties tabs.

B. Compile

To compile, right click on the program name in the RSE Perspective. Click on Compile and then on CRTBNDRPG. You may get a prompt asking you if you want to save your code. If you do, click on OK. This will submit the compile on ZEUS. Wait a few minutes and **Error List will be displayed**

A program will not compile if the error message severity level is greater than 10. It is possible to have level 0 severity messages which are informational and level 10 severity messages which are warnings. The error may

be with your program code syntax or it may be caused by references to objects that do not exist or objects you do not have proper authority to access.

C. Fix your errors.

If you have any errors – fix them and compile again. Errors show on the Error List tab and error messages can be included with your code. The program below had some errors.

The screenshot shows the IBM i development environment. The top pane displays the source code for STUDENTS.RPGLE, which is a one-line program: `...FFilename++IPEASFRlen+LKlen+AIDevice+.Keywords++++++Comments++++++`. The bottom pane shows the Error List tab with the following errors:

ID	Message	Severity	Line	Location
RN9308	Compilation stopped. Severity 40 errors found in program.	50	0	DW233A40/QRPGLESRC(STUDENTS)
RN72319	Not authorized to external description for specified file; specification ignored.	40	1	DW233A40/QRPGLESRC(STUDENTS)
RN7430	Input specifications must be specified for file STUDACCTS.	40	1	DW233A40/QRPGLESRC(STUDENTS)

This one line program is referring to a file (STUDACCTS) that is not found in the your Connection library list. We can easily remedy this. In the Remote systems view, right click on Library List and Type in BCI433LIB as an additional library to be searched for the STUDACCTS object when you are compiling your STUDENTS program using your connection.

The screenshot shows the 'Add Library List Entry' dialog box. The 'Additional library' field is set to 'BCI433LIB'. The 'Where to position additional library' section has 'Start of list' selected. The 'Reference library' field is set to 'QGPL'. The 'ADDLIB' command is displayed as: `ADDLIB LIB(BCI433LIB) POSITION(*FIRST)`.

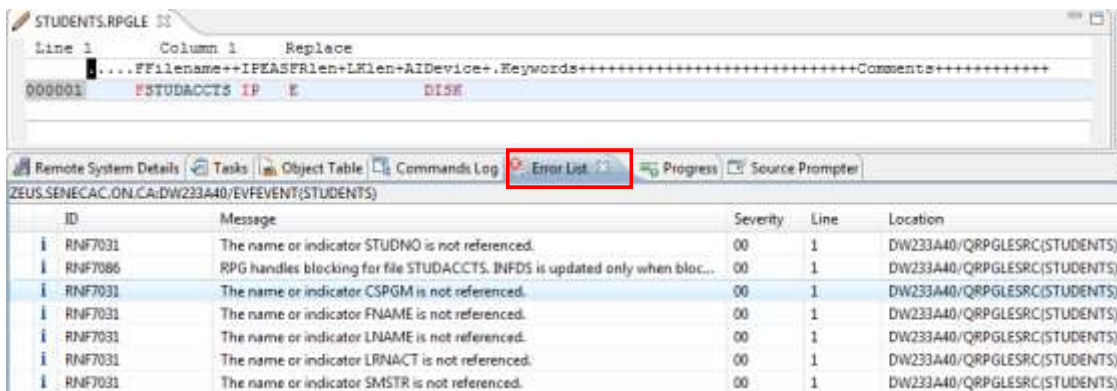
What is the actual CL command that places BCI433LIB on to your list of libraries to be searched when you compile your program in your current library?

`ADDLIB BCI433 POSITION(*FIRST)`

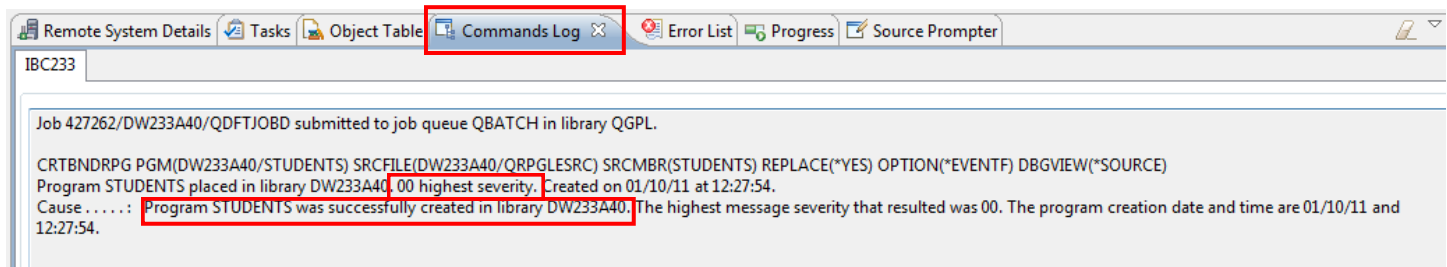
Try recompiling your program with the new library list entry being available. You should get a successful compile.

What is it about the following error list and Commands log that lets you know the compile was successful?

The warnings below are informing us that our one line RPGLE program is not referencing any of the fields in the STUDACCTS file.



The **warnings** are informational and not a good enough reason to prevent the compile and program object creation

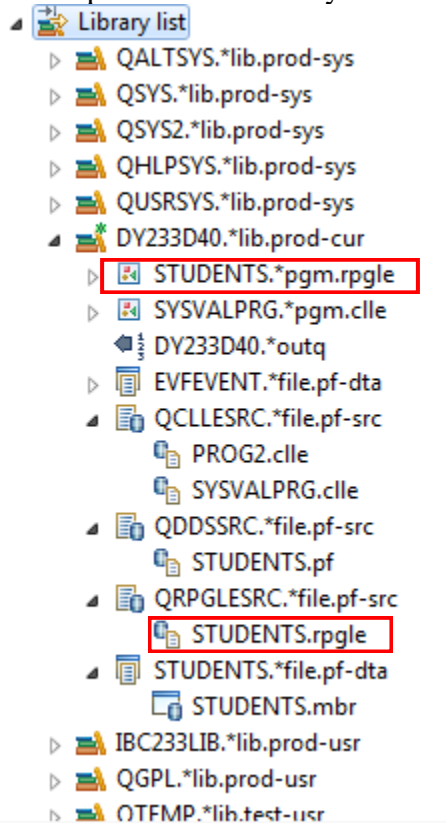


The new program object may not show up in the Remote Systems view. You can get this to appear by right **clicking on the current library** and **selecting Refresh**.

Remember that if you cannot find something in the Remote Systems view, first try the refresh option to ensure you are looking at the latest snapshot of what is on the system.

Double click on the Systems Explorer tab.

You should see something similar to the next screenshot. Use the screenshot provided to identify the named elements.



On the adjacent screen shot see if you can circle the following and draw a connecting line to the following text:

The **RPGLE code** for the compiled object called STUDENTS

The **RPGLE compiled program** object called STUDENTS

The library that wasn't initially part of your library list

The object used to contain the spooled file that was generated when you compiled your program today

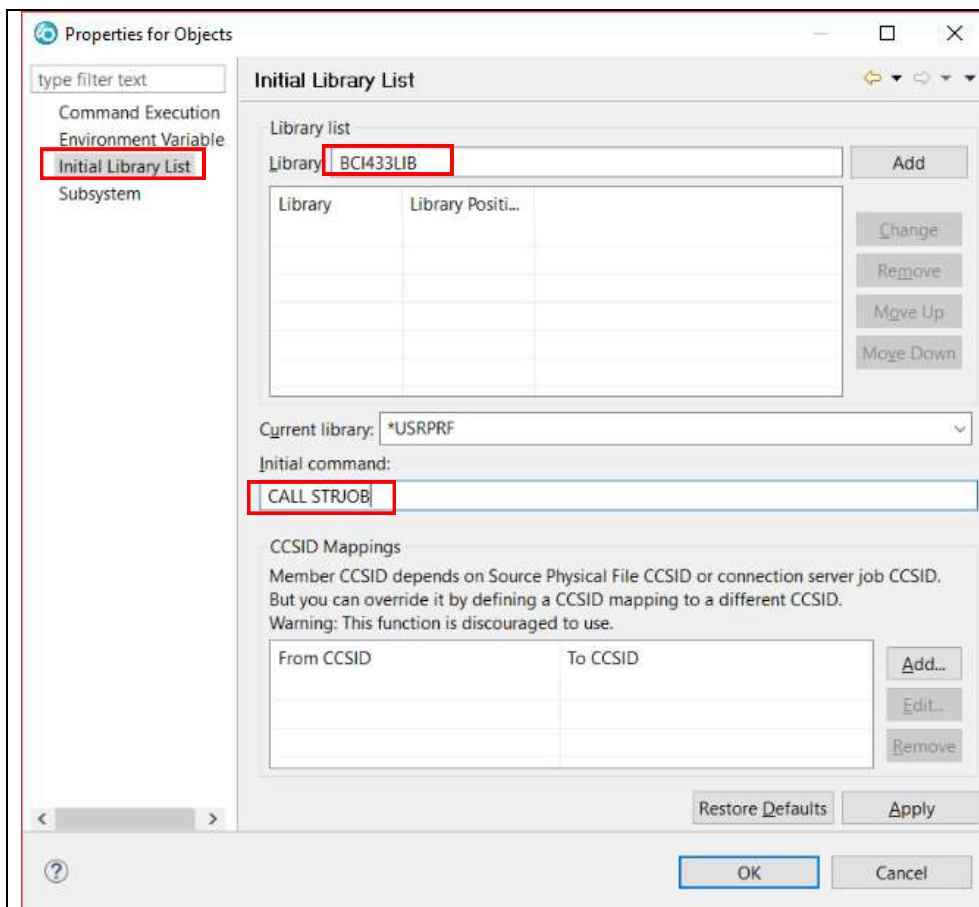
Two CLLE programs have been coded. Which one has not been compiled?

We won't run this program that doesn't do anything.

Exit from RDi and then restart it. Right Click on the Library List for the your connection. Is BCI433LIB included in the list? No (may need to refresh the list)

For those students who are using their computers at home, their laptop in the lab or are saving and restoring their workspaces on a USB there is a way to have BCI433LIB always automatically added to the library list when you initiate the your connection.

Immediately below your connection is **Objects**. Right click on this word and select Properties.



You list of libraries should be blank as currently no additional libraries are being added to your list.

In the right pane you could click on the box beside library. Type BCI433LIB.

The Add button should now be available. Click on it to get this library added to the list below. Click on OK.

Also note under initial Command CALL STRJOB has been entered.

The next time you restart RDi, this library will be included on the list.

You now know how to get a customized library list and have a certain program run for a connection. Remember the lab computers always go back to a default workspace when rebooted.

6. Creating a Physical File.

You have entered CLLE programs using Client Access in lab 1 and entered an RPGLE program using RDi in this lab. Each type of language used its own specialized container or Source Physical File. We are going to create a physical file or table to store data. This can be done a number of ways on the system. Two of the most frequently used methods involve using SQL or entering DDS code and compiling it. We will use the second technique.

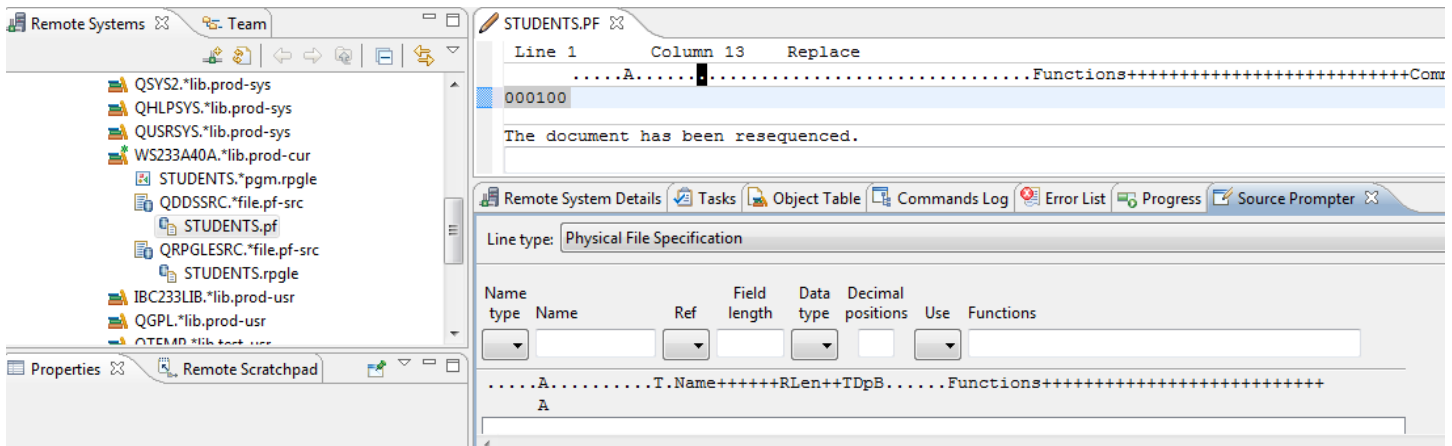
Use RDi to create a container for your DDS code in your library. Call it QDDSSRC and provide a comment indicating that this source physical file object is used to store DDS code.

Create a new member called STUDENTS and indicate the member type is PF (physical file).

ALL THE CODE MUST BE IN CAPITAL LETTERS FOR THIS PROGRAM.

You will see a line in the LPEX editor view that allows prompting. Place your cursor there and press F4. In the Source Prompter view, enter UNIQUE under Functions. Click on the apply tab.

You should see UNIQUE appearing in the LPEX editor and in the Source Prompter.



There are three icons in the upper right of the Source Prompter view. If you hover your mouse over these you will see Disable Source Prompt View, Disable Syntax Checking and Change to Insert Mode.

Click on the last of these icons. Change to **Insert Mode**. or type "I" like green screen

The entry of this program and how to handle syntax errors while entering code will be demonstrated and discussed by your teacher in a lecture period. The following is intended only as an aid to entering the code into your own STUDENTS member.

Enter the following code:

Name	Name	Functions
Type		
R	STUDENTS R	TEXT('STUDENTS REGISTER')

In order to enter the following line you need to realize that **9** is a length, **S** is a Data Type and **0** refers to Decimal Positions

Name	Name	Functions
Type		
	STUDENTNO 9S 0	COLHDG('STUDENT' 'NUMBER')

A first time coder would probably want to continue using the prompter to enter the code.

It is also possible, once you are used to the columns being used by entering the first few lines to double click on the STUDENTS.PF tab and enter the code freeform. You would have to make sure the appropriate entries line up. If you put something in the incorrect column, the live syntax checker will let you know.

Your finished program should look like the following:

Note we are entering an error on purpose here POSTALCODE uses COLDHG instead of COLHDG

```
*STUDENTS.PF X
Line 15      Column 28      Replace  15 changes
.....A.....T.Name+++++.....Functions+++++
000100      A                               UNIQUE
000101      A      R STUDENTSR              TEXT('STUDENT REGISTER')
000102      A      STUDENTNO                9S 0    COLHDG('STUDENT' 'NUMBER')
000103      A      FIRSTNAME                15      COLHDG('FIRST NAME')
000104      A      LASTNAME                 20      COLHDG('LAST NAME')
000105      A      BIRTHDATE                 L
000106      A      PHONE                   10S 0    COLHDG('PHONE' 'NUMBER')
000107      A      ADDRESS                  30      COLHDG('STREET' 'ADDRESS')
000108      A      CITY                    15
000109      A      PROVINCE                 2      COLHDG('PROV' 'CODE')
000110      A
000111      A      POSTALCODE                6      COLDHG('POSTAL' 'CODE')
000112      A      FEESOWED                 7P 2    COLHDG('FEES' 'OWED')
000113      A      FINESOWED                7P 2    COLHDG('FINES' 'OWED')
000114      A      K STUDENTNO
The document has been resequenced.
```

This is just uncompiled code. The STUDENTS physical file will not be created until you successfully compile your code. Remember we have already compiled an RPGLE program called STUDENTS in our student library. You can have several objects with the same name in your library as long as they are a different type of object. You can not have two objects with the same name and object type in your library.

Compile this program using the technique demonstrated earlier with the RPGLE program. You are looking for a CReaTe command here. (CRT)

What was the command to compile this type of member? CRTPF

You may have made a mistake and gotten a compile error. If you followed my code, there is an intentional error. Actually this error wasn't intentional – we all make errors – let's say it after it was discovered, it was intentionally left in.

When you have an error, you can click on the error message in the error list and we should be placed below the coding error. Maybe you already spotted it and fixed it before compiling.

What was the error? 000111 COLDHG()

After you view the error, you can type in the correction and press CTRL + F5 to remove the message from your LPEX editor area. A screenshot shows on the next page with my error. You may have additional ones for this first compile. Fix them and recompile.

Line	Column	Replace
000106	A	PHONE 10S 0 COLHDG('PHONE' 'NUMBER')
000107	A	ADDRESS 30 COLHDG('STREET' 'ADDRESS')
000108	A	CITY 15 COLHDG('PROV' 'CODE')
000109	A	PROVINCE 2 COLHDG('POSTAL' 'CODE')
000110	A	POSTALCODE 6 COLHDG('FEES' 'OWED')
000111	A	FEESOWED 7P 2 COLHDG('FINES' 'OWED')
000112	A	FINESOWED 7P 2
000113	A	

ID	Message	Severity	Line
CPD7484	Keyword not valid for this file type.	20	111

How can you find this new object **STUDENTS.*file** **pf-dta** in the remote systems view?

Do it.

7. Entering Data Using **DFU**.

Switch to your Client Access Session and enter **UPDDTA** STUDENTS at the command line.

Session A - [24 x 80]

File Edit View Communication Actions Window Help

WORK WITH DATA IN A FILE
Format : STUDENTSR

Mode : ENTRY
File : STUDENTS

STUDENT NUMBER: _____
FIRST NAME: _____
LAST NAME: _____
BIRTHDATE: 0001-01-01
PHONE NUMBER: _____
STREET ADDRESS: _____
CITY: _____
PROV CODE: _____
POSTAL CODE: _____
FEES OWED: _____
FINES OWED: _____

F3=Exit F5=Refresh F6=Select format
F9=Insert F10=Entry F11=Change

(C) COPYRIGHT IBM CORP. 1980, 2005.

MA a MW 04/018

1902 - Session successfully started

You are going to enter two records. One record will have your correct first and last name, a 123456789 student number, fees owed of \$100.00 and fines owed of \$20.00. Make up the rest of the data without divulging personal information. The second record will have your instructors first and last name, a student number of 987654321 with 0 Fees and Fines owing. (be kind on the birthdate)

The Data File Utility is a great way to get data into a physical file or table. But, when you use the UPDDTA command without any setup, there are some things to be aware of.

Numeric fields should have valid numeric input. The decimal point doesn't show, so when entering \$94.99 you would enter 9499 and press the field exit key. If you enter 0 in a numeric field without pressing field exit key you will probably get an entry error. The 0 needs to be right justified.

A date field requires a valid date so 2001-02-29 would be rejected.

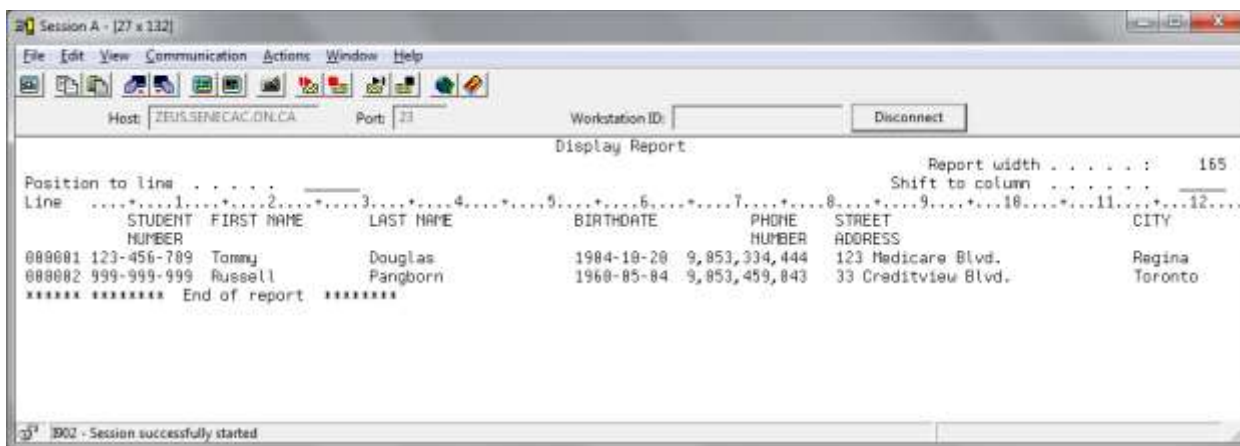
F10 is used to be in **Entry** Mode

F11 is used to be in **Change** Mode

In order to view the data in the file after you have finished with DFU, you can run the following command:

==>**RUNQRY *N STUDENTS**

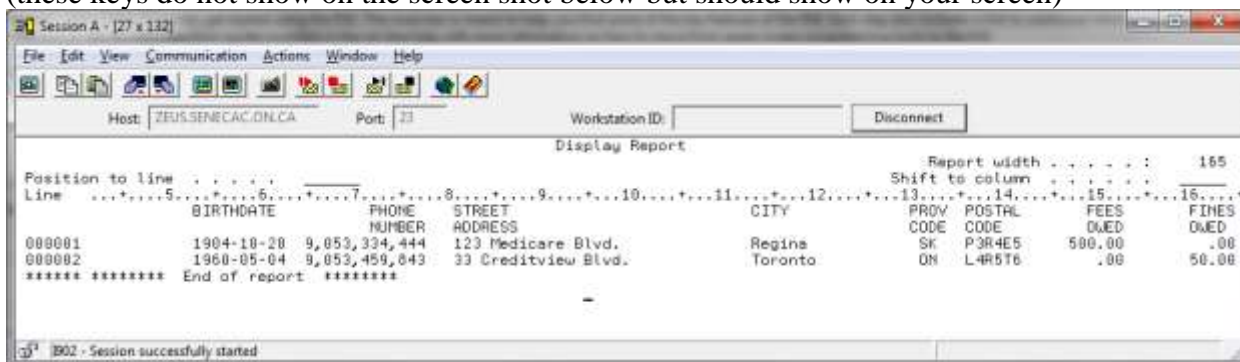
You should get output similar to the following:



The screenshot shows a terminal window titled "Session A - [27 x 132]". The menu bar includes File, Edit, View, Communication, Actions, Window, and Help. The status bar at the bottom indicates "1902 - Session successfully started". The main display area shows a report titled "Display Report" with a width of 165. The report lists student data with columns: STUDENT NUMBER, FIRST NAME, LAST NAME, BIRTHDATE, PHONE NUMBER, STREET ADDRESS, and CITY. Two students are listed: Tommy Douglas (123-456-789) and Russell Pangborn (999-999-999). The report ends with "***** End of report *****".

Line	STUDENT NUMBER	FIRST NAME	LAST NAME	BIRTHDATE	PHONE NUMBER	STREET ADDRESS	CITY
000001	123-456-789	Tommy	Douglas	1984-10-20	9,853,334,444	123 Medicare Blvd.	Regina
000002	999-999-999	Russell	Pangborn	1960-05-04	9,853,459,043	33 Creditview Blvd.	Toronto

What function keys will allow you to see the fees and fines owing for each student? F19, F20
(these keys do not show on the screen shot below but should show on your screen)



The screenshot shows a terminal window titled "Session A - [27 x 132]". The menu bar includes File, Edit, View, Communication, Actions, Window, and Help. The status bar at the bottom indicates "1902 - Session successfully started". The main display area shows a report titled "Display Report" with a width of 165. The report lists student data with columns: BIRTHDATE, PHONE NUMBER, STREET ADDRESS, CITY, PROV CODE, POSTAL CODE, FEES OWED, and FINES OWED. Two students are listed: Tommy Douglas (123-456-789) and Russell Pangborn (999-999-999). The report ends with "***** End of report *****".

Line	BIRTHDATE	PHONE NUMBER	STREET ADDRESS	CITY	PROV CODE	POSTAL CODE	FEES OWED	FINES OWED
000001	1984-10-20	9,853,334,444	123 Medicare Blvd.	Regina	SK	S4R4E5	500.00	.00
000002	1960-05-04	9,853,459,043	33 Creditview Blvd.	Toronto	ON	L4R5T6	.00	50.00

You can also use SQL to view data. The SQL implementation on the System i is very easy to use.

Press Enter to Exit Query/400.

At the command line, type **STRSQL** and press Enter. You are now in the SQL environment. **F4** is your best friend here as it will allow you to prompt any SQL command.

Type **SELECT** and **press F4**. F4 is a widely used OS/400 key that prompts for parameters. Fill in the fields as follows:

FROM files: STUDENTS

SELECT fields: With the cursor in this input field, press F4.
Type numbers (10,20,30 ...) to select and sequence all the fields

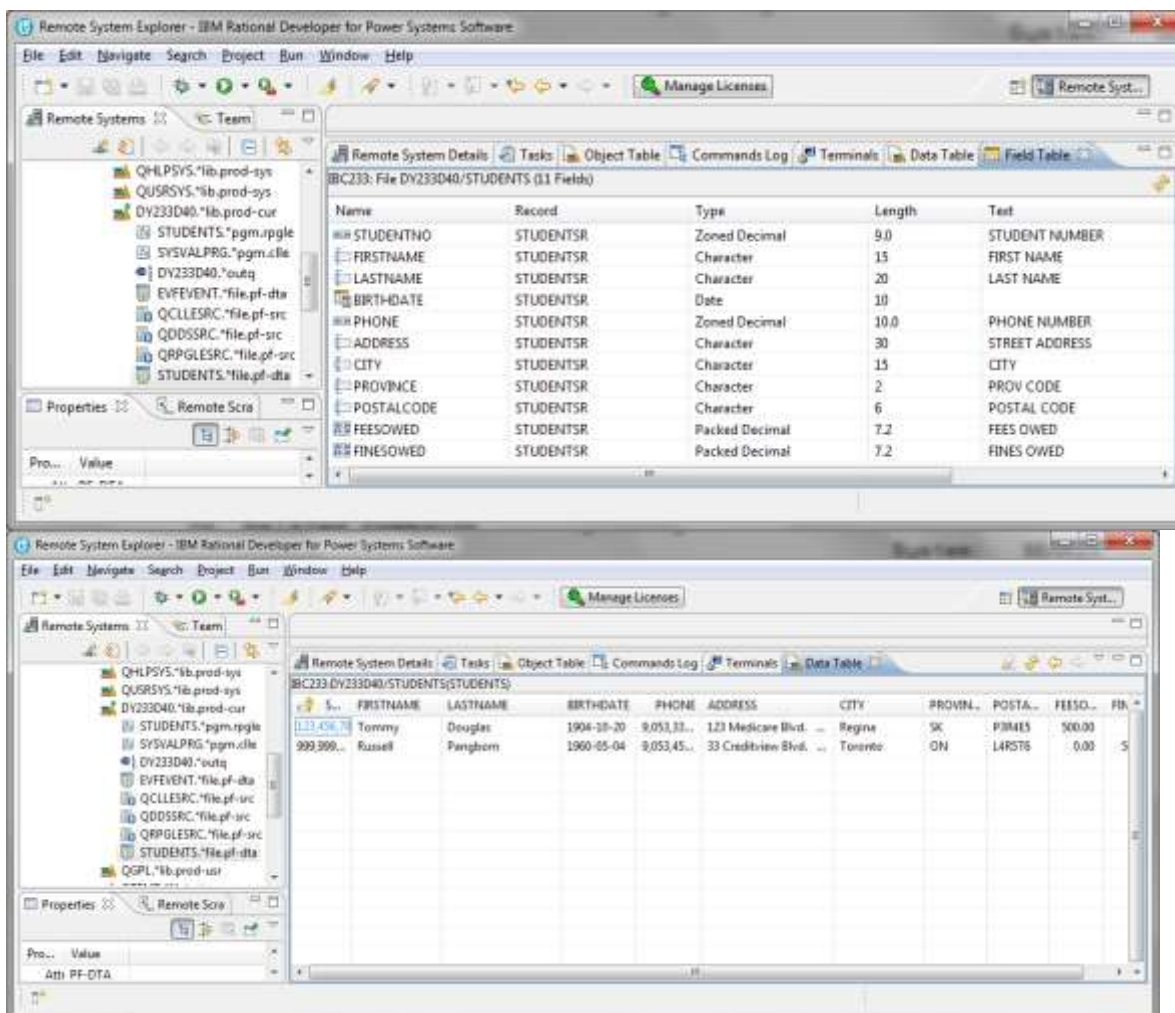
Press Enter twice

The SQL interface allows you more flexibility than the Runqry command. It allows you to select the fields, the order and the records you are interested in.

Press Enter after viewing the file's contents. Press F3 to exit the SQL environment and accept the default option on the "Exit Interactive SQL" screen.

Please return to RDI. We are going to view the data and the field definitions using this tool.

You have a number of things named students in the Remote Systems view. If you right click on the correct students reference and select show in table you can see the field definitions and then show the actual data as shown below. Try it.



Lab 2 Summary (Important notes)

1. RDi uses a workspace that is stored locally on a PC. You can personalize this workspace adding connections to the IBM i system and develop CLLE and RPGLE programs. It is best to save your workspace on a flash drive since the Seneca computers are not set up to keep your workspace changes.
2. Library lists and initial commands can be set up differently for each connection and each connection can use the same or a different user profile.
3. RDi perspectives can be set up to provide different views. The standard set up has a remote systems view on the left side that supports navigation of the system, a work area in the middle that supports developing code or working with a GUI to develop code, a view at the bottom that supports error listings, a commands log that reports the success or failure of attempted commands (compiling) and where a show in table command will place results and in the lower corner a properties view that can show additional information on a highlighted entity like a field.
4. Steps to set up a connection for BCI433
 - Click on IBMi in the remote systems view
 - Host Name: zeus.senecac.on.ca
 - Connection Name: BCI433
 - In the newly created BCI433 connection
 - Right Click on Objects
 - Select Properties
 - On the Properties for Objects screen, click on Initial Library List
 - Type in BCI433LIB in the Library box and click on Add
 - In the Initial Commands Box, Type Call STRJOB
(to have your name appear on listings)
 - Click on OK
5. To see where your workspace is located on the PC click on File(top right) and select Switch Workspace and select Other. You will see the workspace that you need to copy to your flash drive. Click on Cancel, because you do not want to switch your workspace.
6. When you compile a program using RDi and that program refers to a file that is located in a library that is not on your RDi connection library list, the program will not compile successfully.
7. You can fix this by right clicking on Library List in the Remote Systems view and running the Add Library List command every time you restart your connection or by using the strategy outlined in step 4 regarding a permanent solution.
8. A data file (table) can be created by entering DDS code in a member and successfully compiling that code. The usual place for creating this member is in a source physical file called QDDSSRC.
9. Once the file has been created in Client Access you can enter data records with the UPDDTA (update data) command.
10. After you have placed data in the file, you can view this data with the "RUNQRY *N Filename" command in Client Access.
11. In RDi you can view data in a physical file with the Show in Table option. Right click on a file with a *.file.pf-dta extension in the Remote Systems window and select Show in Table and select Data.
12. We can trick you on a test question. Students*pgm.rpgle is not a data file – it is a compiled program. Students.pf inside of QDDSSRC is not a data file with student names and addresses – it is source code used to produce the data file.
13. You can use CREATE TABLE instead of entering DDS code and compiling it when producing a table/data file/physical file.