**BCI433 – Lab 2**

Summer May 2022

Objectives:

* Introduction to RDi (IBM Rational Developer for i) and Eclipse
* Create and successfully compile an RPGLE program
* Print out a compiler 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

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. (Access Client Solutions)

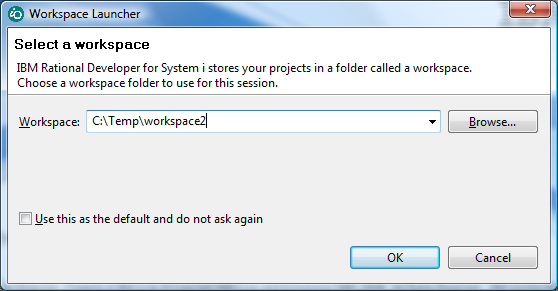
We’ll be using RDi in this lab. RDi stores the development environment on your PC. The default workspace is a folder in the temp directory.

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 i.** Your instructor has demonstrated how this product can be started.

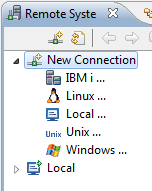
Depending upon how your computer is configured, you may be prompted for your default workspace:



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. |

1. **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.Senecacollege.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. |

1. **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 dialouge shows it is a good  idea to check off Save User ID.  It is not a good idea to check off Save password |

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

**2-Sign on** to the iSeries using the Access Client emulation tool and determine the objects you have in your library. ===> WRKOBJPDM DS233snn (substitute your library name here)

Object Type Attribute

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.**

**For the DS233snn object there are the following objects with the same name:**

**Object type are:** **Libraries containing objects:**

3 - 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?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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.

|  |  |
| --- | --- |
| **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? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  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?  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  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.

5 - 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 Access Client sessions and also have a different library lists in each session by adding or removing libraries from the list with CL commands.

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

**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 IBM 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)  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. |

**6. Creating a Physical File.**

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.

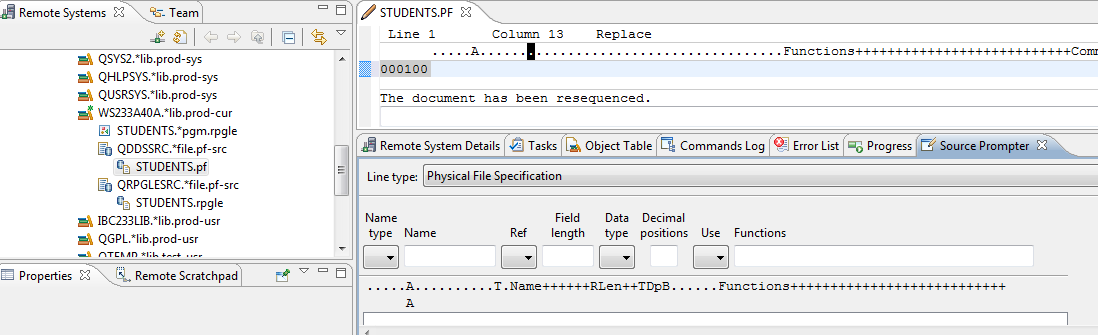
Create a new member called STUDENTS and indicate the member type is PF (physical file).

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.

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 STUDENTSR 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

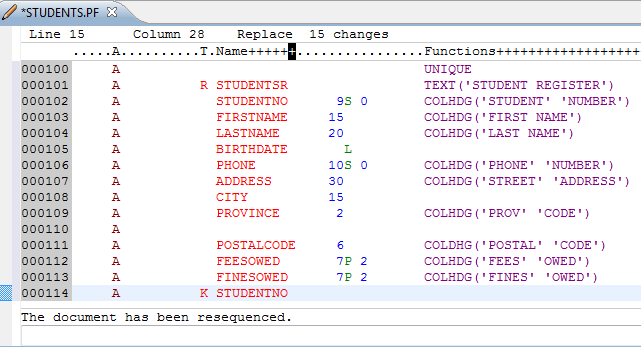
STUDENTNO 9S 0 COLHDG('STUDENT' 'NUMBER')

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



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)

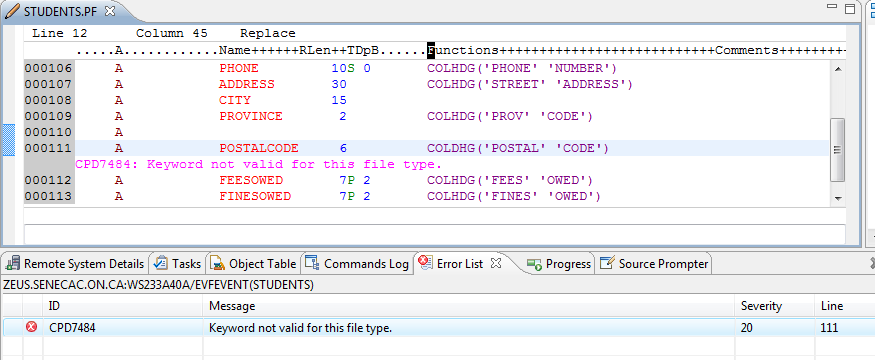
What was the command to compile this type of member? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

You may have made a mistake and gotten a compile error. If you follow 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? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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.



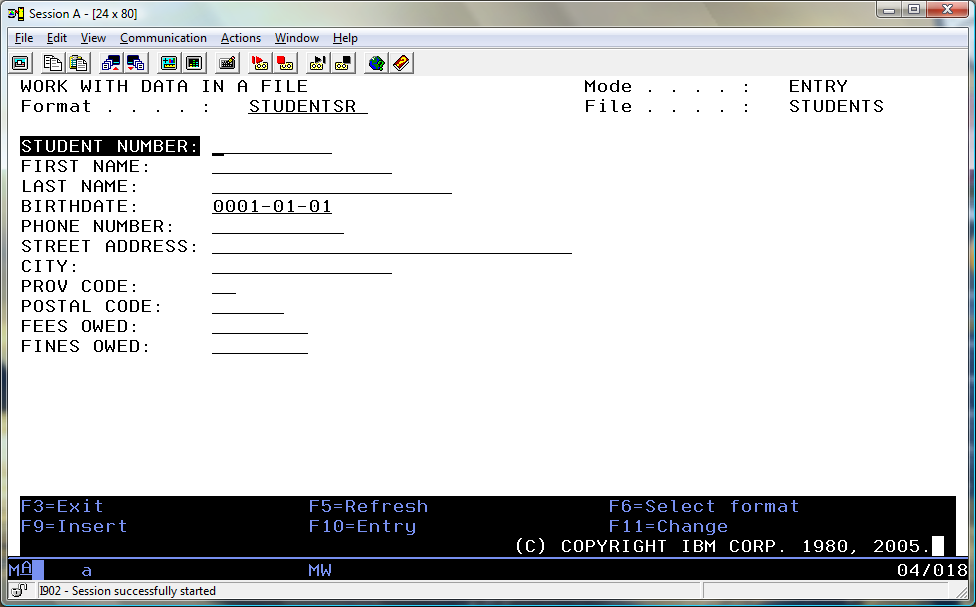
How can you find this new object STUDENTS.\*file.pf-dta in the remote systems view?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Do it.

**7. Entering Data Using DFU Data File Utility**

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



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 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.

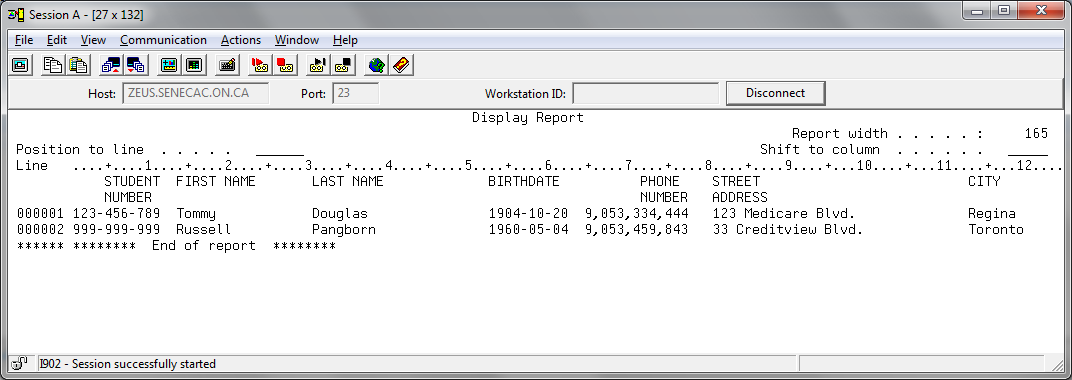
F10 is used to be in Entry Mode

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:

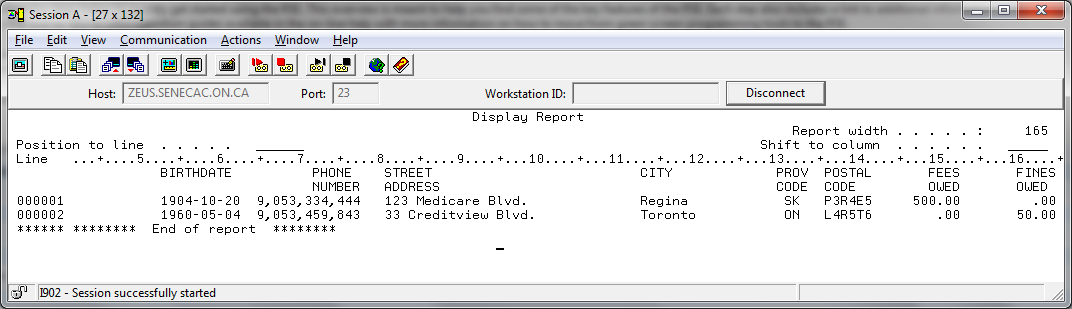
==>RUNQRY \*N STUDENTS

You should get output similar to the following:



What function keys will allow you to see the fees and fines owing for each student? \_\_\_\_\_\_\_\_\_\_\_\_

(these keys do not show on the screen shot below but should show on your screen)



**You can also use SQL to view data. The SQL implementation on the IBM 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.

