Lab objectives:

Create an RPGLE program that determines employee pay using an externally described printer file developed using Report Designer.

Run a file override from the command line to have your RPGLE program produce different results

Requirements to pass the lab:

Successfully run the RPGLE Payroll program with and without a file override.

Report Designer

The following report layout will be designed to be used in a later lab by an RPGLE program:

```
3/01/2017  14:24:55                  Weekly Payroll Report                      Page:0001
 Name: Student Name

   Employee      Work      Pay     Hourly    Hours    Regular        Overtime           Total
    Number      Shift     Grade     Rate     Worked     Pay             Pay              Pay

  333-333-333     D         1       16.00      41      664.00         24.00             688.00

  122-222-222     D         3       14.20      35      497.00                           497.00

  322-222-222     D         2       15.70      40      628.00                           628.00

 .  .  .


                          Totals:          $28,685.69      $499.68      $29,185.37
```

This externally compiled printer file will be available to programs of all languages that print reports on the system.

The programmer will not have to provide the layout for the report. Instead they will just refer to the various types of output records available to them. They will write Title, ColHdg, EmpDetail and Totals.

Hopefully when you look at the report above, you can pick out those items.

The RDi GUI tool Report Designer will be used to develop your externally described printer file report.

Here is the code you will not be entering:

```
A               R TITLE
A                                  1 45'Weekly Payroll Report'
A                                  1 89'Page:'
A                                  1 94PAGNBR
A                                  1  3DATE(*YY)
A                                     EDTCDE(Y)
A                                  1 15TIME
A               R COLHDG              SPACEB(1)
A                                   6'Employee'
A                                     SPACEB(1)
A                                  19'Work'
A                                  29'Pay'
A                                  37'Hourly'
A                                  47'Hours'
A                                  56'Regular'
A                                  71'Overtime'
A                                  89'Total'
A                                   7'Number'
A                                     SPACEB(1)
A                                  19'Shift'
A                                  29'Grade'
A                                  38'Rate'
A                                  47'Worked'
A                                  58'Pay'
A                                  74'Pay'
A                                  90'Pay'
A               R EMPDETAIL           SPACEB(2)
A                 EMPNUM     R   9S 0  5REFFLD(ALLSHIFT/EMPNUM SENECAPAY/A-
A                                     LLSHIFT)
A                                     EDTWRD('0   -   -     ')
A                 WORKSHIFT R    1A  20REFFLD(ALLSHIFT/WORKSHIFT +
A                                     SENECAPAY/ALLSHIFT)
A                 PAYGRADE   R       30REFFLD(ALLSHIFT/PAYGRADE +
A                                     SENECAPAY/ALLSHIFT)
A                 HOURLYRATE     5  2 37EDTCDE(1)
A                 HRSWORKED R        48REFFLD(ALLSHIFT/HRSWORKED +
A                                     SENECAPAY/ALLSHIFT)
A                 REGULARPAY     7  2 55EDTCDE(1)
A                 OVERPAY        7  2 70EDTCDE(1)
A                 TOTALPAY       7  2 86EDTCDE(1)
A               R TOTALS             SPACEB(2)
A                                  41'Totals:'
A                 TOTREGPAY      9  2 51
A                                     EDTCDE(1 $)
A                 TOTOVTPAY      9  2 66EDTCDE(1 $)
A                 TOTEMPPAY      9  2 82EDTCDE(1 $)
```

Instead of entering code, open up RDi and create a source physical file called PayrollApp.

Create a new member in PayRollApp called PAYRPT and use the type PRTF (printer file)

Close this member and reopen it with Report Designer. Use the same technique that use used with Screen Designer to do this.

An absolute record is required first. You can create this by dragging Absolute Record from the left side Palette and releasing it in the work area. The default name for this is RECORD1. You can change this record name to TITLE in the Properties view on the lower right.
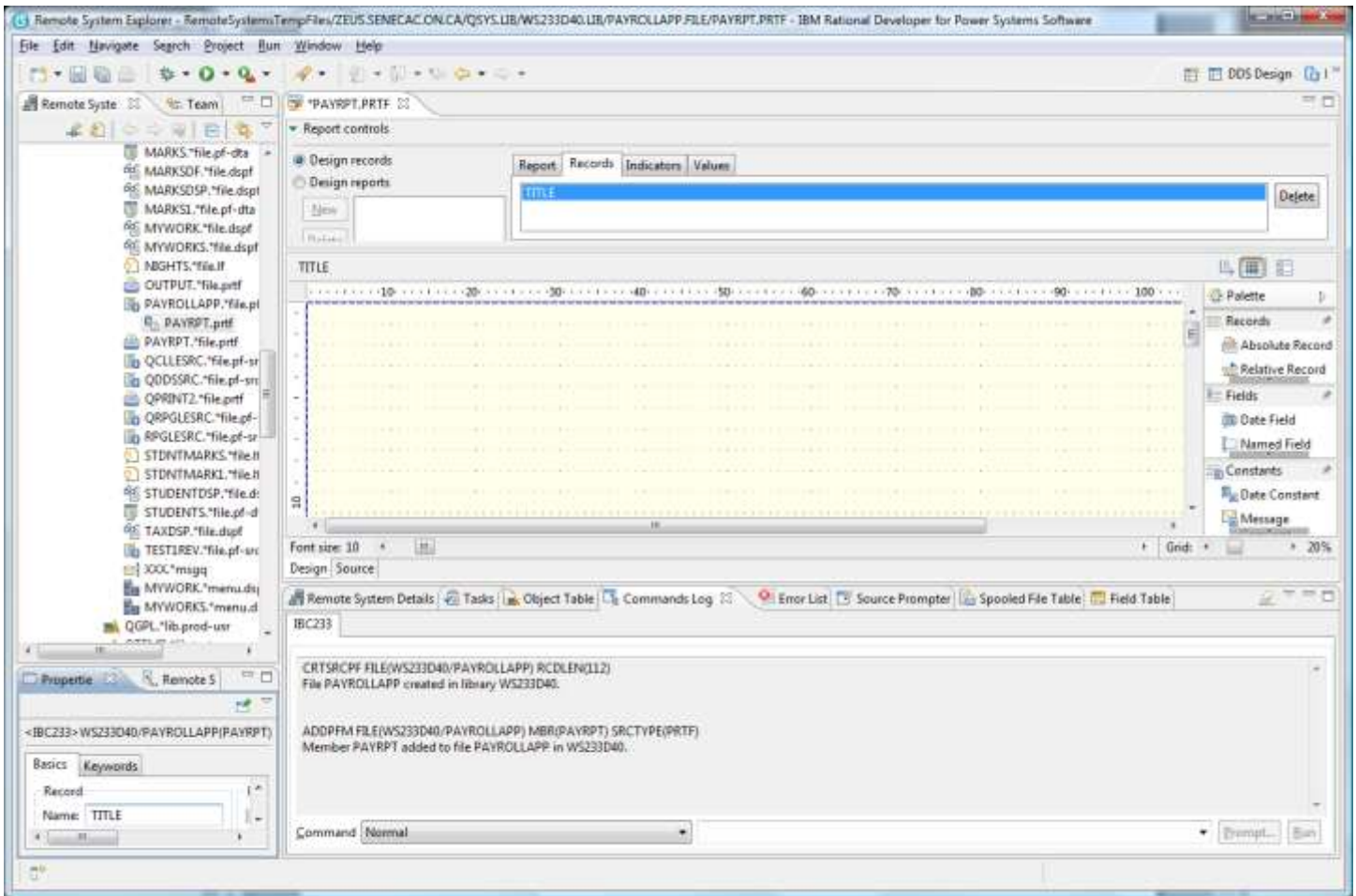

Double click on the PAYRPT.PRTF tab to get an expanded view of the palette. The content that needs to be added to the TITLE record includes a Date Constant, Time Constant, Text Constants and a Page Number Constant. All of this is found in the palette. You do not have to place everything in the exact column number that this report uses, but make it all visually fit between column 1 and column 100. Also include on the second line a Name label and your actual full name.

As you are working on this report, you can press CTRL+S occasionally to save your work.

There are Design and Source tabs available so you can look at the DDS code that is being generated.

The finished product shows on the next page for TITLE .
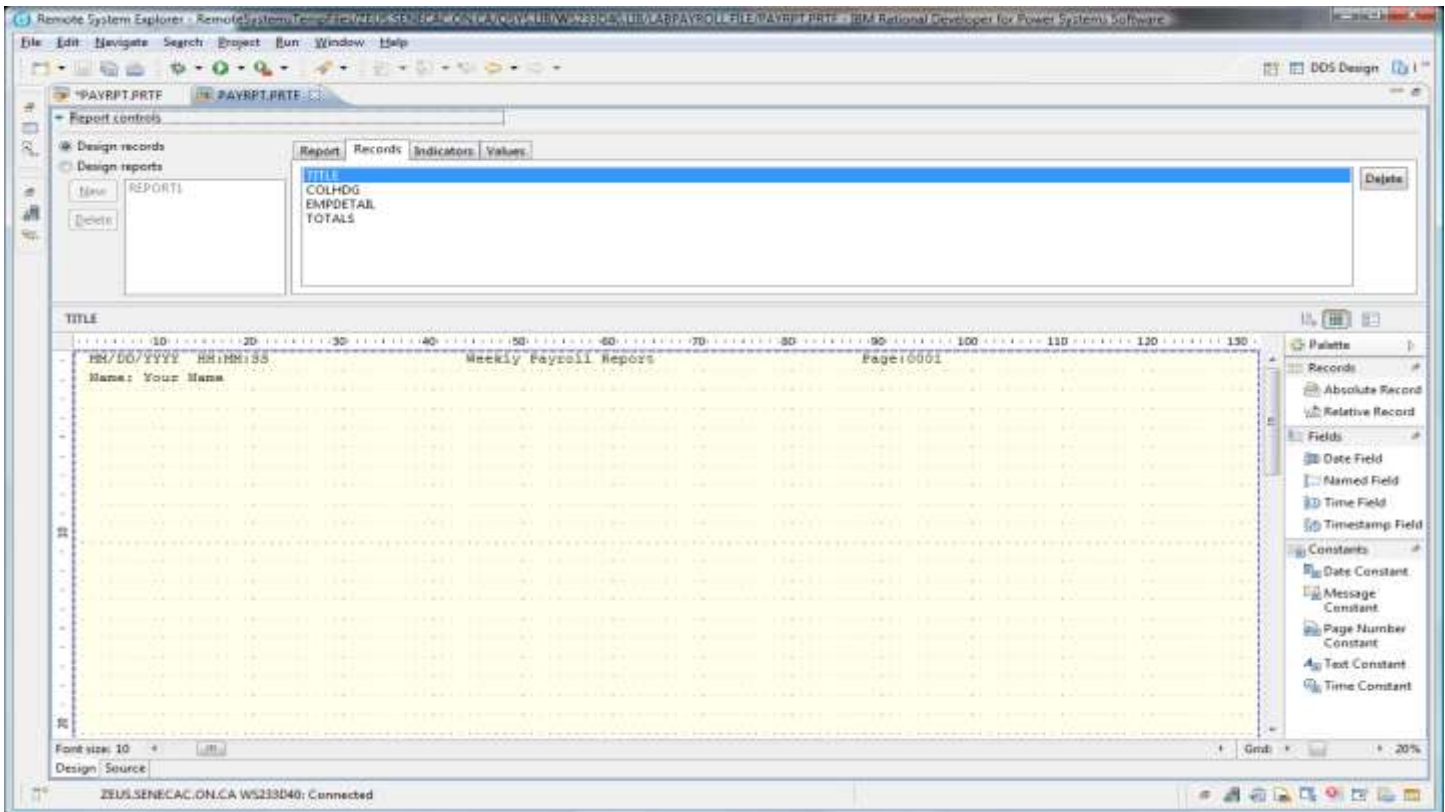
The next record you add from the palette should be a Relative Record called COLHDG. Drag this over and make sure you drop it on line 3 or 4. TITLE does not show, but you will be able to line everything up later when Designing Reports.
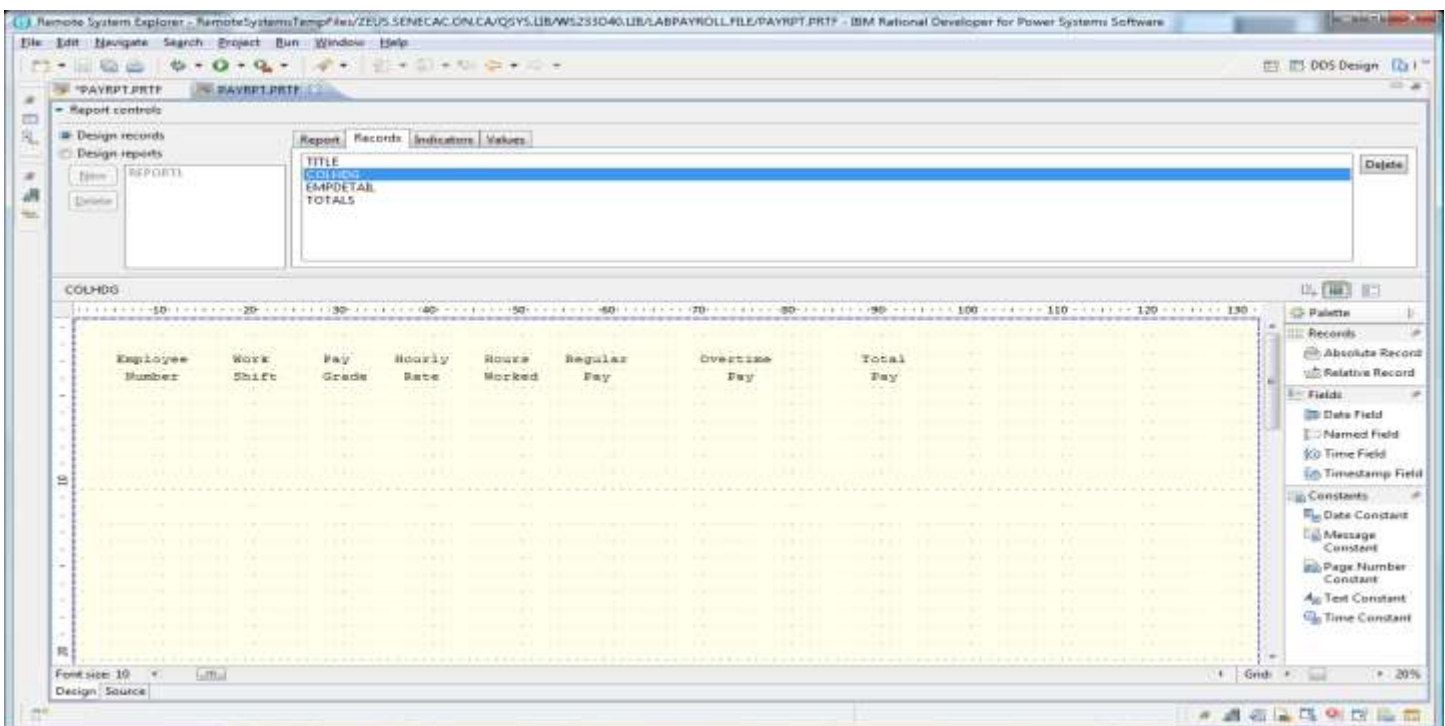
The EMPDETAIL relative record will have the fields from a file called ALLSHIFT in the SENECAPAY library dragged onto the screen on a line number lower than the column headings.

Notice in the screenshot below an object filter was created to the SENECAPAY library and the ALLSHIFT object. Then the ALLSHIFT.*file.pf-dta was right clicked on with a Show in Table – Fields selected for the field table.

Each field was dragged to an appropriate area on the screen and then the properties view was used to supply edit codes or edit words. An edit code of "2" was used on all the numeric fields except for the employee number. An edit word was used for the employee number. **Create named fields for HOURLYRATE, REGULARPAY, OVERPAY, and TOTALPAY.**
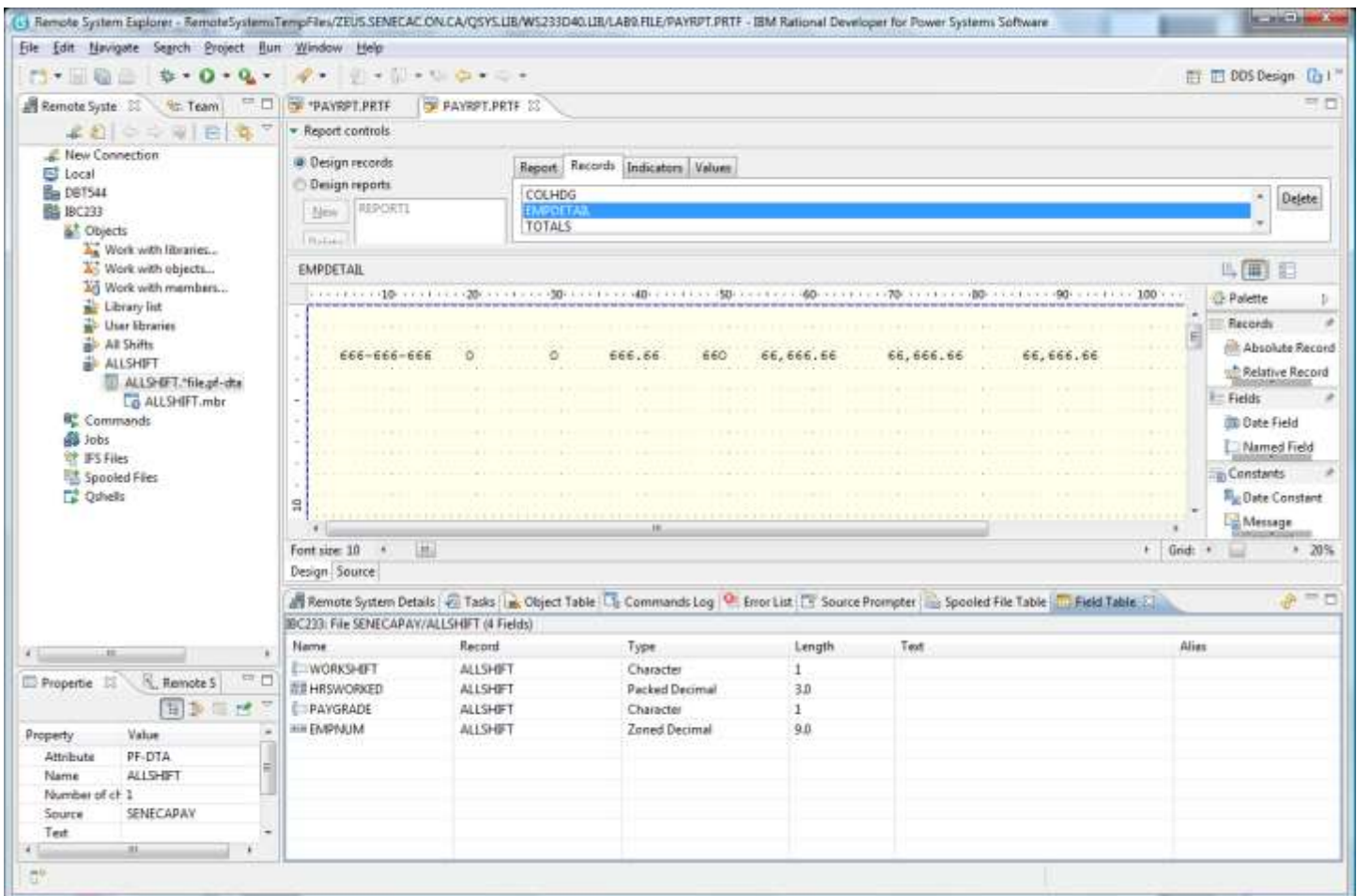

Provide a Totals record with **named fields** used for the last three column totals. TOTREGPAY, TOTOVTPAY and TOTEMPPAY. These fields are zoned numeric and 9 digits with 2 decimal places. These field also use an edit code of "1" and include the currency symbol.

You are now ready to put all this together. There is a report control area under the PAYRPT.PRTF tab at the upper right. It was set at Design records for this first stage.

Click on the Design reports radio button and click on the Report tab to change the name from untitled to Report1 and click on the Apply button.
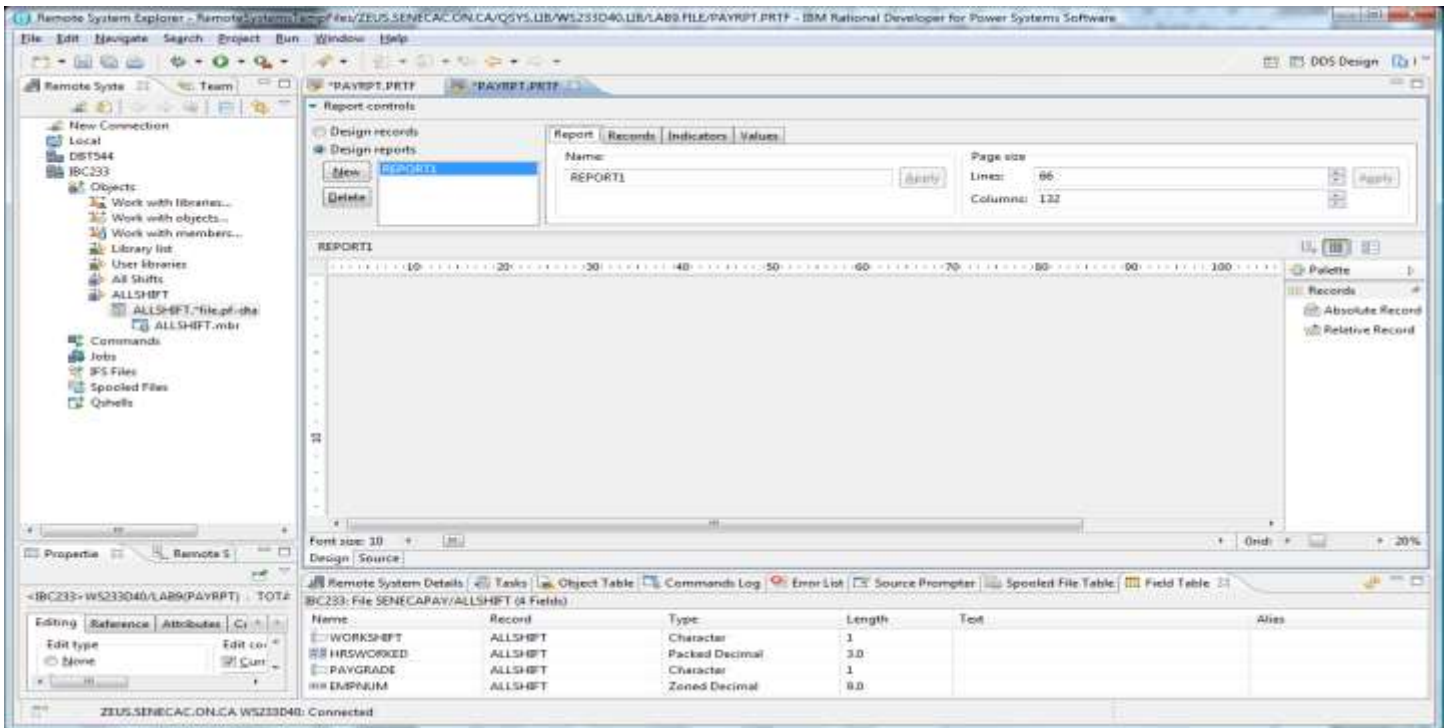
Click on the Records tab and use the Add button to add one record at a time in the proper order.
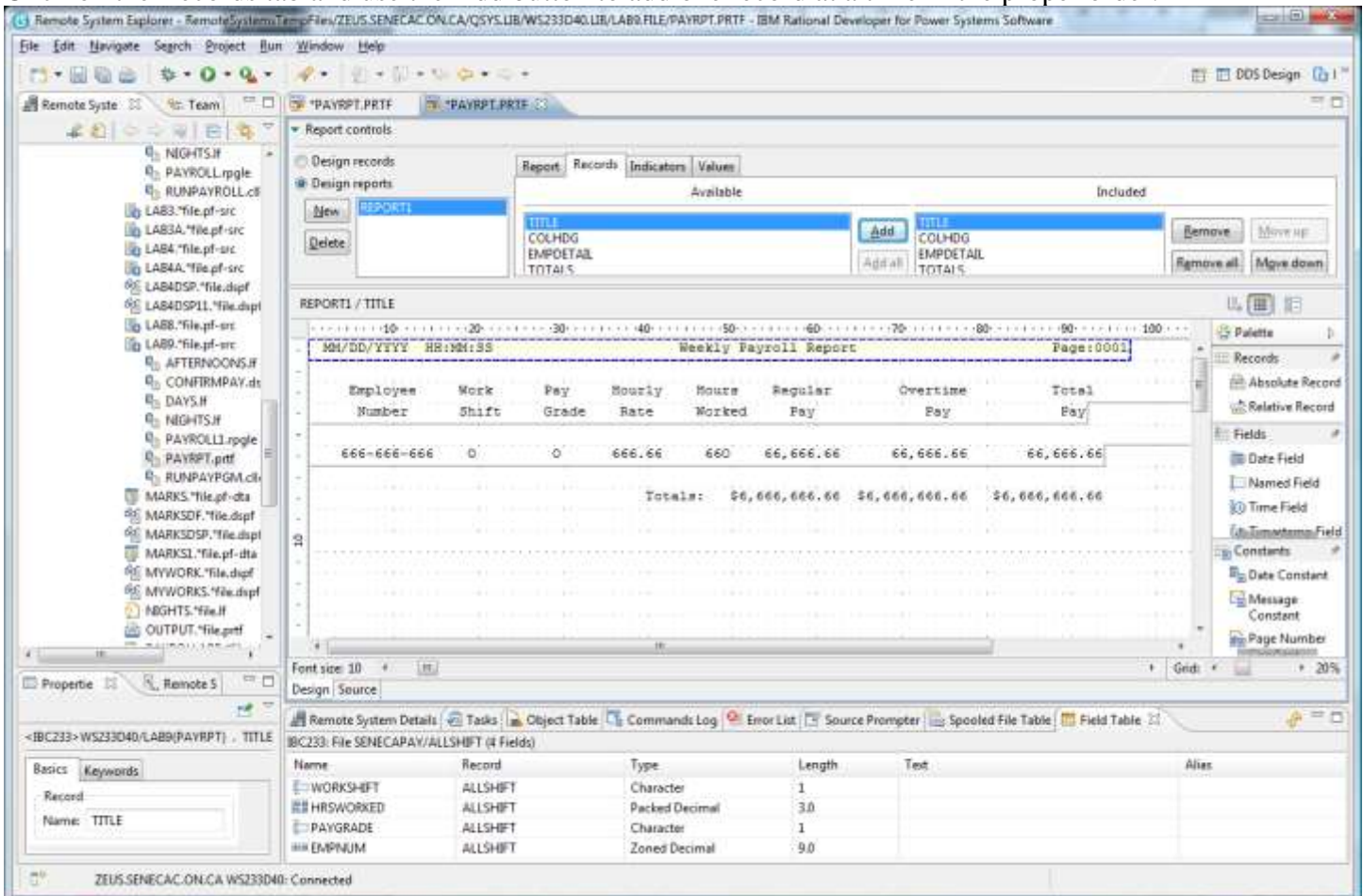
When you have added all the records, you can adjust where the fields line up or where the column headings line up by clicking on the record content.

After you have moved everything around so it appears similar to the last screenshot shown, save your work so it can be viewed by an instructor.

Your last step is to compile the DDS code that has been generated for you.

Click on the SOURCE tab in the DESIGN/SORUCE area and then using the top menu compile option with the CRTPTF command.

A successful compile produces an externally described file in your library that can be used by programs.

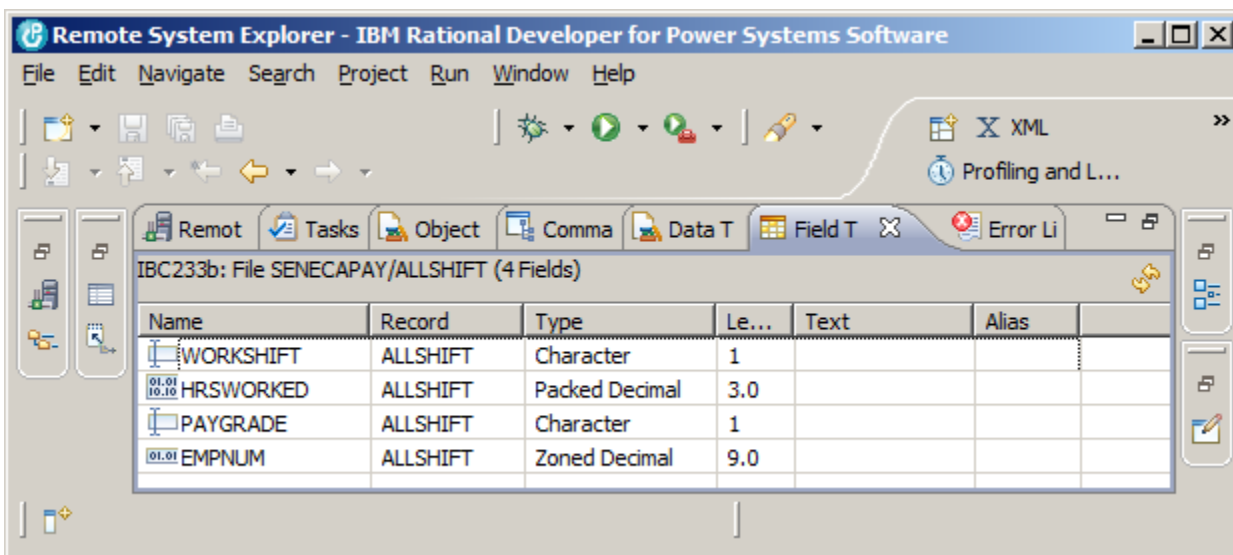| Object	Type	Attribute
PAYRPT	*FILE	PRTF

Any programmer can refer to this file in their program and they are relieved of entering the code that describes how the report will appear.

RPGLE

INPUT

Add the SENECAPAY library to your library list to allow you to pick up externally described fields from the SHIFTRATES and ALLSHIFT files for your RPGLE program.

Use the Show in Table feature to look at the data in the files and the field definitions.

What is the field information for the ShiftRate Table?

| Name	Record	Type	Length
DAYRATE	ShiftRates	Zoned Decimal	5.2
NIGHTRATE	ShiftRates	Zoned Decimal	5.2
AFTNRATE	ShiftRates	Zoned Decimal	5.2

Program

The program you are coding should produce similar results to an RPGLE program called PayrollPgm.
This program is found in BCI433LIB and that should already be part of your library list if your initial program is working. The data is found in the SENECAPAY library. You can either run a command to always include this library in your library list every time you sign in, or make an adjustment to your initial program when running the program in Client Access.

Your RDi workspace requires the same choice – make this an activity you have to do every time you start RDi or something that automatically happens so the compile step will be able to find the externally described files in SENECAPAY.

Run the instructor provided program in Client Access.

A screen record from a display file should show:
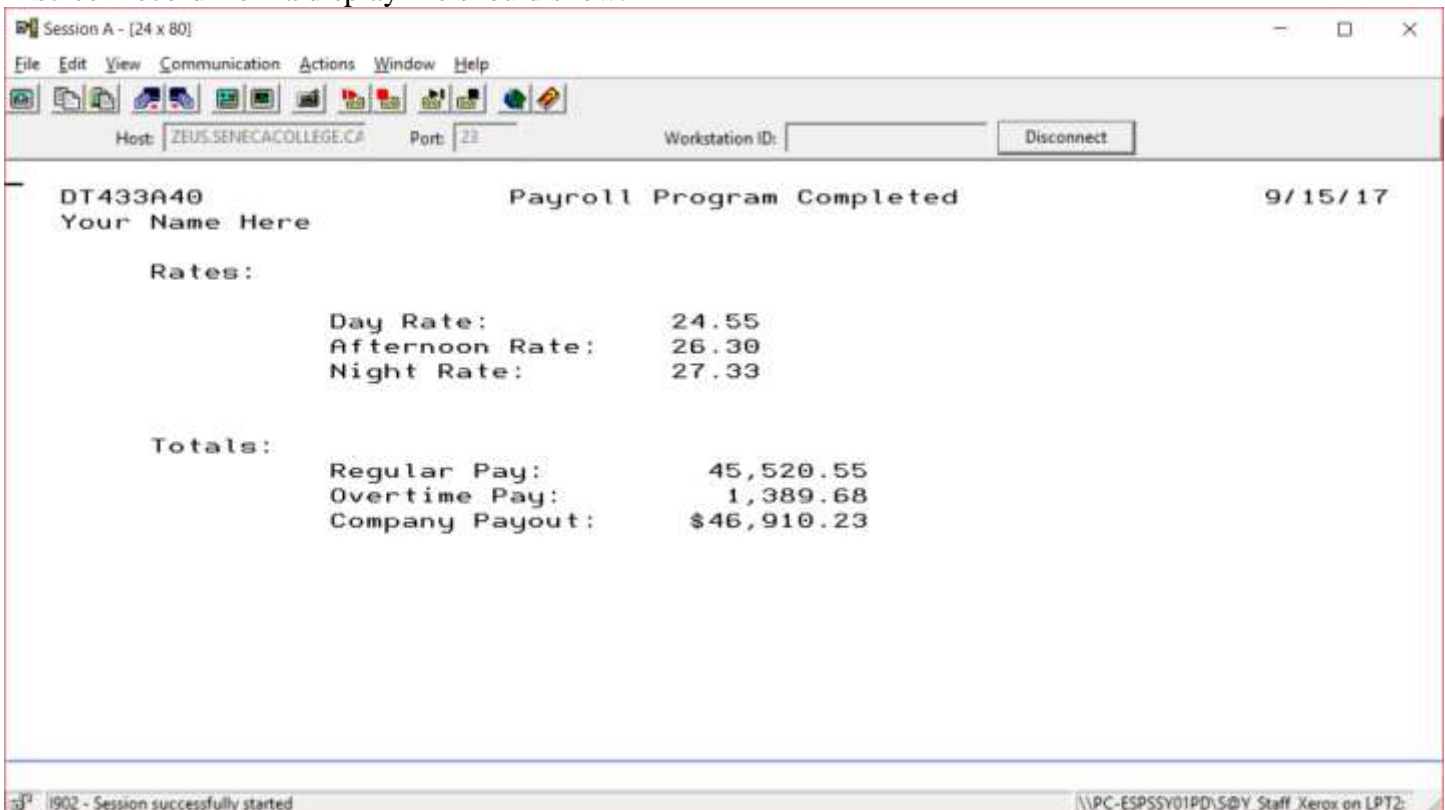
Note that your name in place of "Your Name Here" will identify your work when demonstrating the finished program.

Take a look at the output for the program in your output queue. The numbers will be different, but it should look similar to the following:

```
  3/11/2017  14:05:51        Weekly Payroll Report            Page:0001

 Name: Your Name

 Employee      Work    Pay     Hourly   Hours   Regular   Overtime    Total
  Number       Shift   Grade    Rate    Worked    Pay       Pay        Pay

 333-333-333    D       1      17.31     41     692.40     25.97     718.37

 122-222-222    D       3      15.15     35     530.25       .00     530.25

 322-222-222    D       2      16.67     40     666.80       .00     666.80
 ...

 567-567-567    A       3      15.91     39     620.49       .00     620.49

                               Totals:       $?????????   $?????? $???????
```

Your name will be included on your report.

You have already developed the externally described printer file that will be used by your RPGLE program.

All you need to do is determine which output lines get printed at specific times and how to handle overflow printing; the logic for calculating pay, overtime pay and total pay; and handle reading the records from the file.

Printing:

You have a printer record called Title. When writing this to a report, you will automatically get to a new page. This is useful at the start of the report so it does not print on the same page as an unrelated report and is useful when you want to get to a new page after reaching the overflow line while your report is printing.

Write Title; is the RPGLE code you need to use.

What are the other output record names supported by your externally described printer file?

for the column headings: _____   for the detail line: _____

for the summary line   _____

In order to handle overflow you need to select an indicator from 01 to 99 and identify it as an overflow indicator to the program.

FFilename++IPEASF.....L.....A.Device+.**Keywords**+

FPAYRPT   O   E   PRINTER   **OFLIND(*IN01)**

This overflow indicator will be automatically turned on for you when you print a detail line on the overflow line. Usually this is line 60 on a 66 line page. This can be reset to a different line number with the OVRPRTF command.

The logic dealing with overflow can appear just before you code to print a detail line.

Basically you need to check the indicator.   If it is on, print the Title output record and any other needed heading records and then ensure the indicator is turned off. (The system automatically turns this indicator on and the programmer is required to turn it off.)

Payroll Processing:

The logic to determine the amounts paid for regular pay, overtime pay, total weekly pay for the employee and totals for all employees can be handled in a subroutine. Determine a good time to invoke this subroutine in your main routine and call this subroutine PaySr.

In this subroutine you will be adjusting the HourlyRate (a field initially defined on a display file record and brought into your program at compile time.)

The first adjustment is based on one record that is found in a file called SHIFTRATES.
You should get this record at the start of your program, before you enter the loop that processes all the AllShift records.

Your WorkShift field found in the ALLSHIFT file can be a "D", "A", or "N" for Day, Afternoon or Night shift times.

HourlyRate will either be the DayRate, NightRate or AftnRate.

The second adjustment is based on Paygrade from the AllShift file:

    Paygrade   1 - 7.3% higher rate than the rate for their shift
                2 - 5.4% higher rate than the rate for their shift
                3 - 4.5% lower rate than the rate for their shift

After you have determined the HourlyRate, you need to calculate the workers pay based on a 40 hour work week.

Any hours over 40 would have an overtime rate applied which is time and a half to the overtime horus.

Workers working 40 hours or less would only get their rate that you already determined based on shift time and paygrade.

After you have determined the employee pay, appropriate total fields are incremented for all employees.

Definition specifications for calculated totals

Lab 7 asked you to research fields that are work fields. These fields need to be defined and are not part of any externally described files.

Here is a sample Definition Spec, Stand Alone field that is required for the Payroll program:
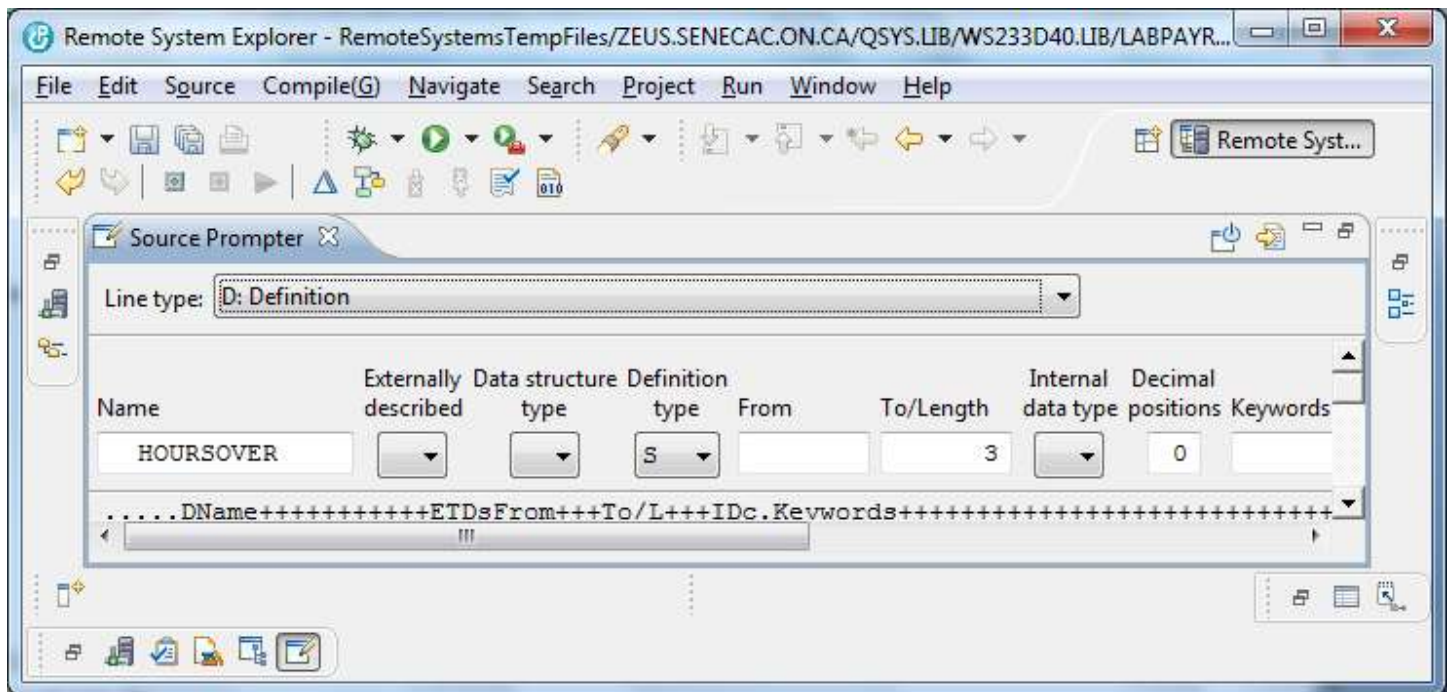
FILES:
Four files need to be declared in your program. Use the display file name CONFIRMPAY.

File Type	FT (IOUC)	Record Length	RL	
File Designation	FD (PSRTF)	Record Address Type	RAT (ADFKPDZ)	
End of File	EF	Device	(Printer,Disk,Workstn)	
File Addition	FA	Keywords		
File Format	FF (FE)			

Filename	FT	FD	EF	FA	FF	RL	RAT	Device	Keywords
F_____	__	__	__	__	__	___	___	_____	_____
F_____	__	__	__	__	__	___	___	_____	_____
F_____	__	__	__	__	__	___	___	_____	_____
F_____	__	__	__	__	__	___	___	_____	_____

Include the HoursOver field

Name	S/U	Declaration Type	From	To/ Length	Data Type	Decimal Pos	Keywords
D___HOURSOVER__	___	____	___	___	___	___	_____

```
/FREE
     // T H I S   C O D E   W I L L   B E   D E V E L O P E D   I N   C L A S S   IF ABSENT – GET NOTES

     //                 M A I N   R O U T I N E
```

```
*INLR = *ON;
RETURN;
```

```
// P A Y S R

 BEGSR PaySr;
```

```
  ENDSR;
```

Produce the CONFIRMPAY display file

It reports on the Day, Afternoon and Night rates read from the ShiftRates data file and Totals determined after reading all the AllShift records.

Compile your RPGLE program.

Run your program and check your spooled file output.

If your program is running properly, you are ready for the final phase of this lab.

File Overrides

The AllShift file contains records for day, afternoon and night shift workers.

You can isolate a single shift type of workers in your report without having to make any changes in your RPGLE program. This powerful feature is available to you by creating a view and using the OVRDBF command.

To create a view on ALLSHIFT in the SENECAPAY collection/library you need to run STRSQL at the command line in Client Access.

In your interactive SQL session:
```
    CREATE   VIEW     DT433D40/NIGHTS AS
             SELECT * FROM SENECAPAY/ALLSHIFT
             WHERE   WORKSHIFT = 'N'
             RCDFMT   ALLSHIFT
```

This view only includes night shift workers. You can compare the entire file with the view.

```
SELECT * FROM SENECAPAY/ALLSHIFT
SELECT * FROM NIGHT
```

In order to get your RPGLE program to only process night shift workers, at the command line enter the following command:

```
OVRDBF ALLSHIFT     NIGHT
```

You are overriding all references to the file ALLSHIFT to be directed to the View NIGHT.

To confirm the override is in effect run the following command:
DSPOVR

If you press F3, you will be at a different invocation level and may lose the overrid
Call your program. It should only be processing the night shift workers.

The view object NIGHTS does not show the results in Employee number order.
(not the amounts and data will not be the same for your report)

```
 11/17/2016  15:52:32          Weekly Payroll Report                    Page:0001
```

Name: Your Name

Employee Number	Work Shift	Pay Grade	Hourly Rate	Hours Worked	Regular Pay	Overtime Pay	Total Pay
444-444-444	N	1	20.07	40	?????		?????
522-222-222	N	2	19.33	40	?????		?????
143-444-433	N	3	17.58	40	?????		?????
243-343-433	N	3	17.76	40	?????		?????

An index can be created to show Employee number order, but we will get all shifts included instead of just the night shift.

A logical file can be used to produce an object similar to a view.

A logical file can also produce an object that can't be created with SQL. We will produce this object in our next lab. This logical file will only show night shift workers and will also present those records in employee number order.

Authorities

What change needs to be made for the *Public in order for students to build views over SencaPay/AllShift?

```
Session A - [24 x 80]                                                    _ □ ×
File  Edit  View  Communication  Actions  Window  Help
 ▣  🗎🗎  🖥🖥  🖳🖳  ▣  🗎🗎  🗎🗎  🗎  🌐🖉
                        Edit Object Authority

Object . . . . . . . . :    ALLSHIFT        Owner  . . . . . . . . :    PANGBORN
  Library . . . . . . :    SENECAPAY       Primary group  . . . :    *NONE
Object type  . . . . :    *FILE           ASP device . . . . . :    *SYSBAS

Type changes to current authorities, press Enter.

   Object secured by authorization list  . . . . . . . . . . . . .    *NONE

                          Object     ----------Object----------
User          Group       Authority  Opr  Mgt  Exist  Alter  Ref
*PUBLIC                   *USE         X    _     _      _     _
PANGBORN                  *ALL         X    X     X      X     X




                                                                     Bottom
F3=Exit    F5=Refresh    F6=Add new users      F10=Grant with reference object
F11=Display data authorities   F12=Cancel   F17=Top   F18=Bottom

MA     a                 MW                                          09/068
 I902 - Session successfully started
```
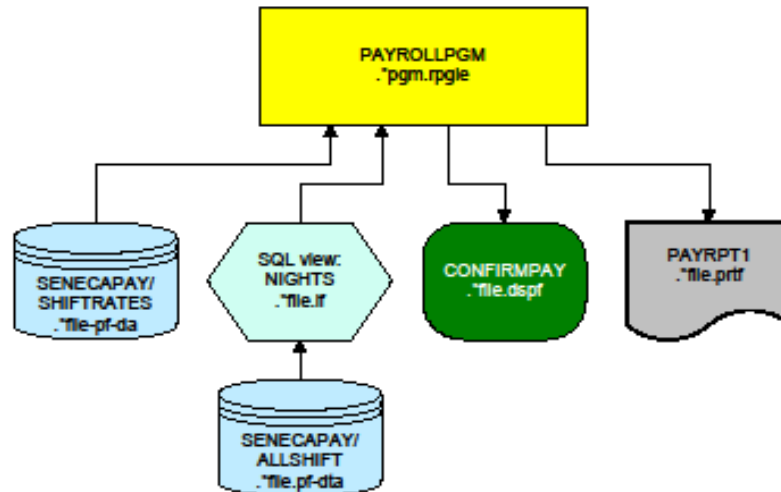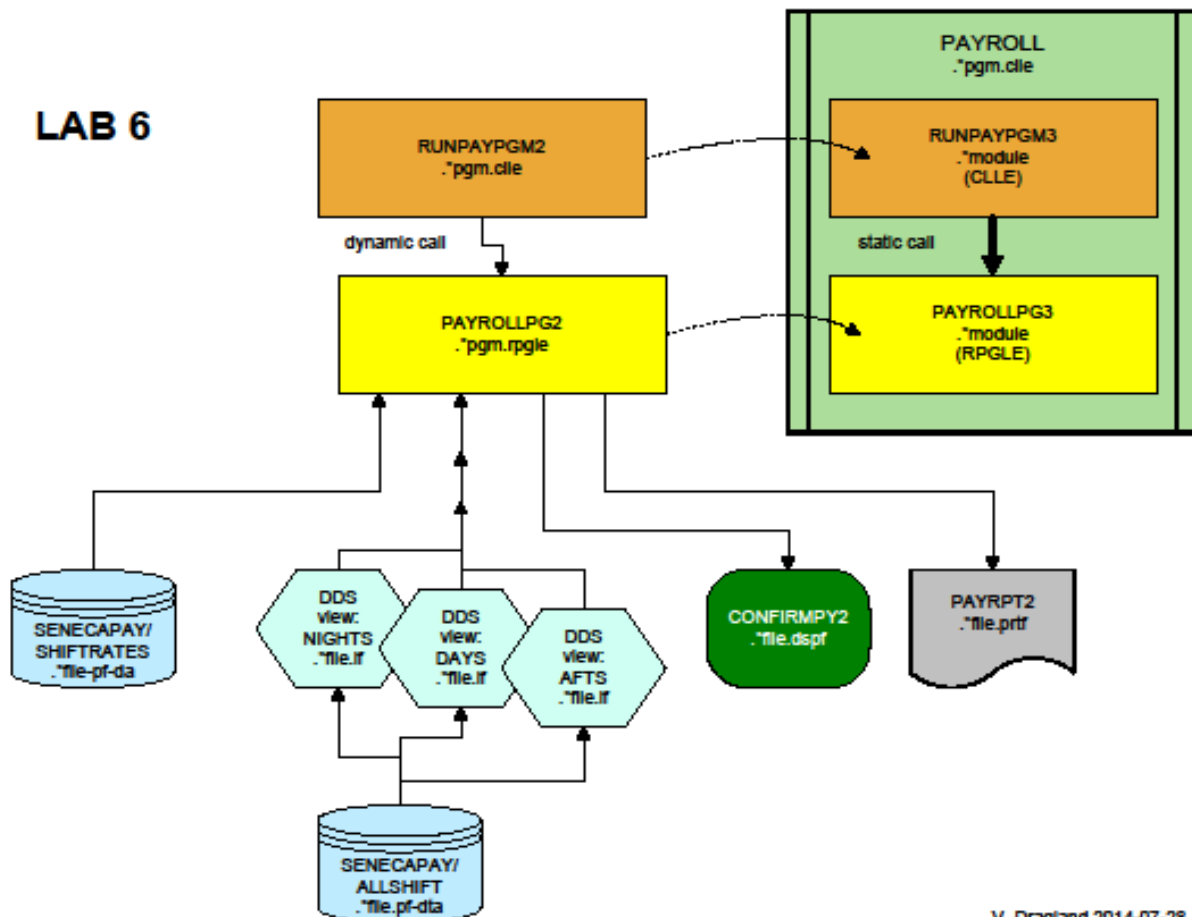
_____

_____

Seneca BCI433

LAB 5

PAYROLLPGM
."pgm.rpgle

SENECAPAY/
SHIFTRATES
."file-pf-da

SQL view:
NIGHTS
."file.lf

CONFIRMPAY
."file.dspf

PAYRPT1
."file.prtf

SENECAPAY/
ALLSHIFT
."file.pf-dta

LAB 6

RUNPAYPGM2
."pgm.clle

PAYROLL
."pgm.clle

RUNPAYPGM3
."module
(CLLE)

dynamic call

PAYROLLPG2
."pgm.rpgle

static call

PAYROLLPG3
."module
(RPGLE)

SENECAPAY/
SHIFTRATES
."file-pf-da

DDS
view:
NIGHTS
."file.lf

DDS
view:
DAYS
."file.lf

DDS
view:
AFTS
."file.lf

CONFIRMPY2
."file.dspf

PAYRPT2
."file.prtf

SENECAPAY/
ALLSHIFT
."file.pf-dta

V. Dragland 2014-07-28