

BCI433 Lab 3 (Summer 2018)

Writing an interactive RPGLE screen program

Lab objectives:

- Create a display file (screen)
- Code an interactive RPGLE program using the display file

Lab Requirements:

Hand in the compile listing for HWYTOLLRPG
Successfully run HWYTOLLRPG

Start an RDi session

Start a 'Green Screen' (emulator) Session.

Using Rational Developer for i (RDi):

Part A

Objectives:

- Use RDi Screen Designer to Create a Display File

Display File screens in IBM i are very similar to forms you see on Web pages. Entering and compiling **DDS code** can produce these interactive screens. One option that may be used for entering DDS code is to use a GUI tool that allows you to visually select and enter what appears on the screen. Your selections are converted to DDS program code that is compiled to produce interactive screen records in a display file object. Those interactive screens are available to various program languages.

We will **create a display file object** called **HWYTOLLDSP** that uses two overlapping screen records. RPGLE, CLLE and COBOL programs can interact with the screens in the HWYTOLLDSP object. We will **code an RPGLE program** called **HWYTOLLRPG** to view and interact with the screen records. The following shows a run of the RPGLE program.

==>CALL HWYTOLLRPG

first screen called **TOLLINFO** is displayed

The screenshot shows a green screen emulator window titled 'Session A - [24 x 80]'. The main display area shows the following text:

```
DM433B40                                1/19/18
                                H I G H W A Y   4 0 8   T O L L S
                                T r i p   I n f o r m a t i o n

Kilometers traveled:  _____
Type of Vehicle:      _ (L - Light, H - Heavy, M - Multiple Heavy)
Time of Day:          _ (01 -12)  _ (AM or PM)
Transponder:         _ (Y N)
```

At the bottom of the window, there is a status bar with the text '1902 - Session successfully started' on the left and 'IPC-ESPSSY01PDI.S@V_Staff_Xenos on LPT2' on the right.

Data is entered and the ENTER key is pressed

A second screen called SHOWTOLL overlays the first screen and the input fields are protected

Session A - [24 x 80]

File Edit View Communication Actions Window Help

Host: ZEUS.SENECACOLLEGE.CA Port: 23 Workstation ID: Disconnect

DM433B40 1/19/18

HIGHWAY 408 TOLLS

Trip Information

Kilometers traveled: 55

Type of Vehicle: L (L - Light, H - Heavy, M - Multiple Heavy)

Time of Day: 8 (01 -12) AM (AM or PM)

Transponder: N (Y N)

Applicable Toll Charges

Toll Rate: \$.3707

Toll Charge: \$20.38

No Transponder Charge: 4.07

Total Charge: 24.45

F3 - Exit

R02 - Session successfully started \\IPC-ESPSSV01PDI3@V_Shell_Xerox on LPT2:

The user can either press F3 to exit the program or enter to check out other usage combinations. F3 is only available for use from SHOWTOLL.

Using Screen Designer to Produce a Display File

Source code for a Display File may be stored in a source file called **QDDSSRC**. You have already created this file in Lab 2 to store DDS code in the STUDENTS member that produced a physical file called STUDENTS.

DDS Stands for Data Description Specifications. The native IBM i language used to describe *FILE objects.

Create a new member inside of QDDSSRC. The member name should be specified as HWYTOLLDSP and the file type should be DSPF. After creating the member, close the HWYTOLLDSP tab.

Right click on your HWYTOLLDSP member, click on Open With and select SCREEN DESIGNER.

You should still see a Remote Systems view on your upper right. The following should appear:

An outline view on the right.

A tab in the centre that has your member name (HWYTOLLDSP) with two boxes below the tab. One box has a title of Screens and the other box includes a Screen and Records tab

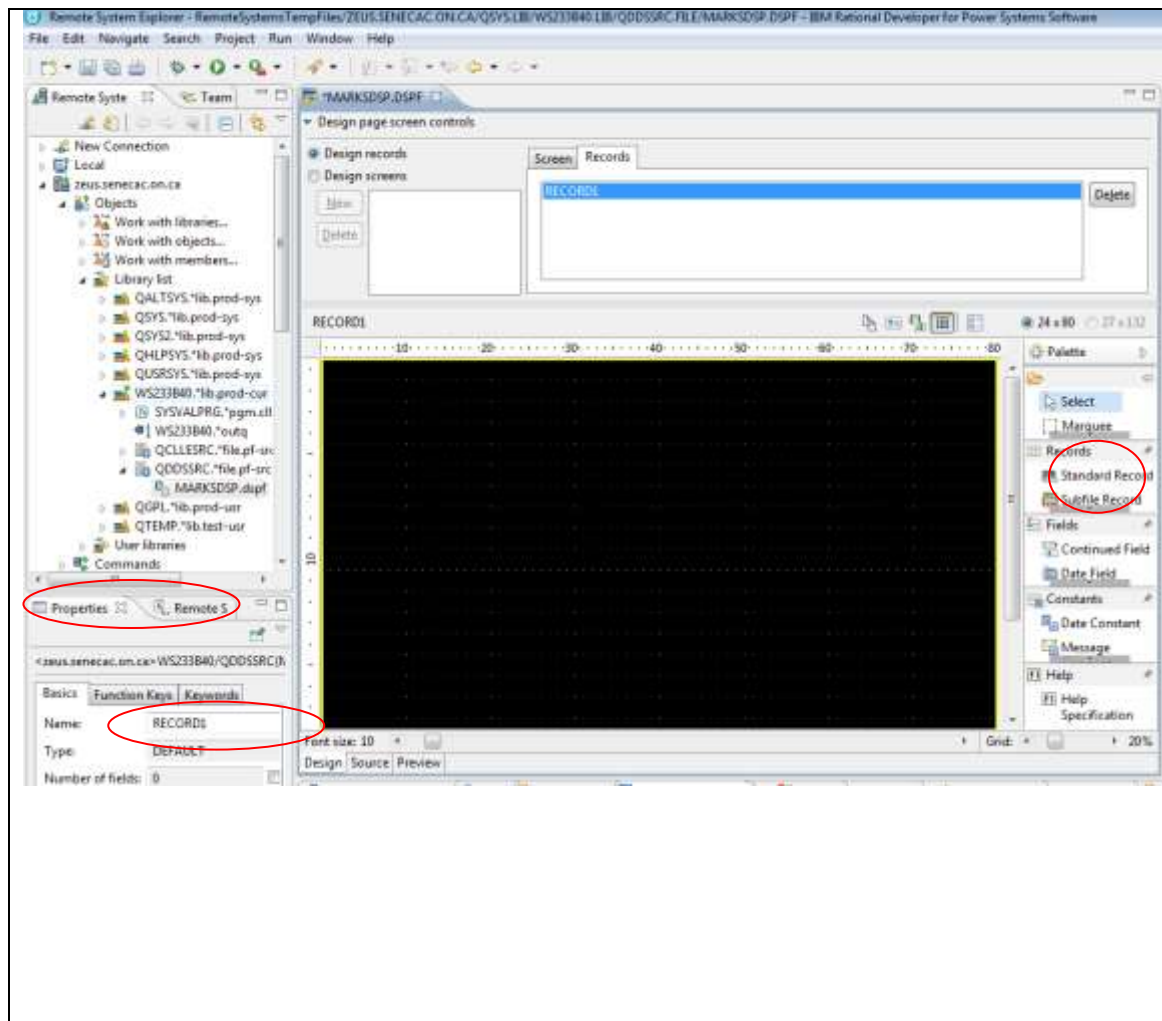
Below this a work area should appear with a ruler that counts from column 1 to column 100

A palette is available to the right of this work area and this will be used to select and place fields and constants on our screen record.

Immediately below this work area will be tabs that allow us to work with the GUI or to work with actual DDS code

Further down is a Source Prompter tab. We have already used a source prompter in lab 2 to enter RPGLE code and to enter DDS code for a physical file program.

A properties box on the lower left which can be used to enter useful information about records and fields.



Left click on **Standard Record** in the Palette area and drag this over to the left work area.

Record1 should show up in several spots. In a right top window and in a **Properties box** on the lower left. This name can be changed to something else here. We will change it to **TOLLINFO**.

Create a date constant (MM/DD/YY) by clicking on Date Constant which is found **in the palette underneath the Constants folder**. Move your cursor over to the top left of your black grid screen and click once to place the date constant. As you move your cursor with the mouse, you should see the row and column numbers changing. After you have placed this date, go back to the constants folder in the palette and use the arrows to scroll through other constant selections. We are using most of them. Refer to the finished first screen on a previous page to determine the placement of these constants and then add them to your work area.

Highway 408 Tolls, Trip Information, and Kilometers Traveled: ... are text constants.

Click on Text Constant in the palette and then click to place this centred at the top of your screen. You can change the default text of "Text constant" to "Highway 408 Tolls" one of two ways. Type directly in the box provided. If this is awkward click on the word "Text Constant" in the properties box in the lower left corner.

Put in all the Text constants using the finished product screenshot.

We need to put in fields beside the test, lab and exam text constants.

Click on **Named Field** below the Fields folder in the palette. Place the named field beside the Kilometers Traveled: constant.

This shows up as **BBBBBBBBBB** on your work screen. Double click on the Properties box to specify information about this field. The field name that was provided for you is FIELD1. This can be easily changed in the box beside name.

Change the field name to KILOMETERS.

The default properties for this field are set as **9 character Input/Output field**. The usage reflects this in Properties with a setting of **"Both"**.

Make this a **3 character** field by typing over the "9" that appears under **Length** on the right side of the properties view.

The **usage** for the field should **remain as Both**. **This means** it is **input capable** (the user can **enter** something into the field when it is presented on the screen record and it is **output capable**, the program or user can **set** the field contents and it will **show** up to the user the next time the screen is displayed.

What appears on your work screen with the following?

Usage set to INPUT |||||||
Usage set to OUTPUT oooooooo

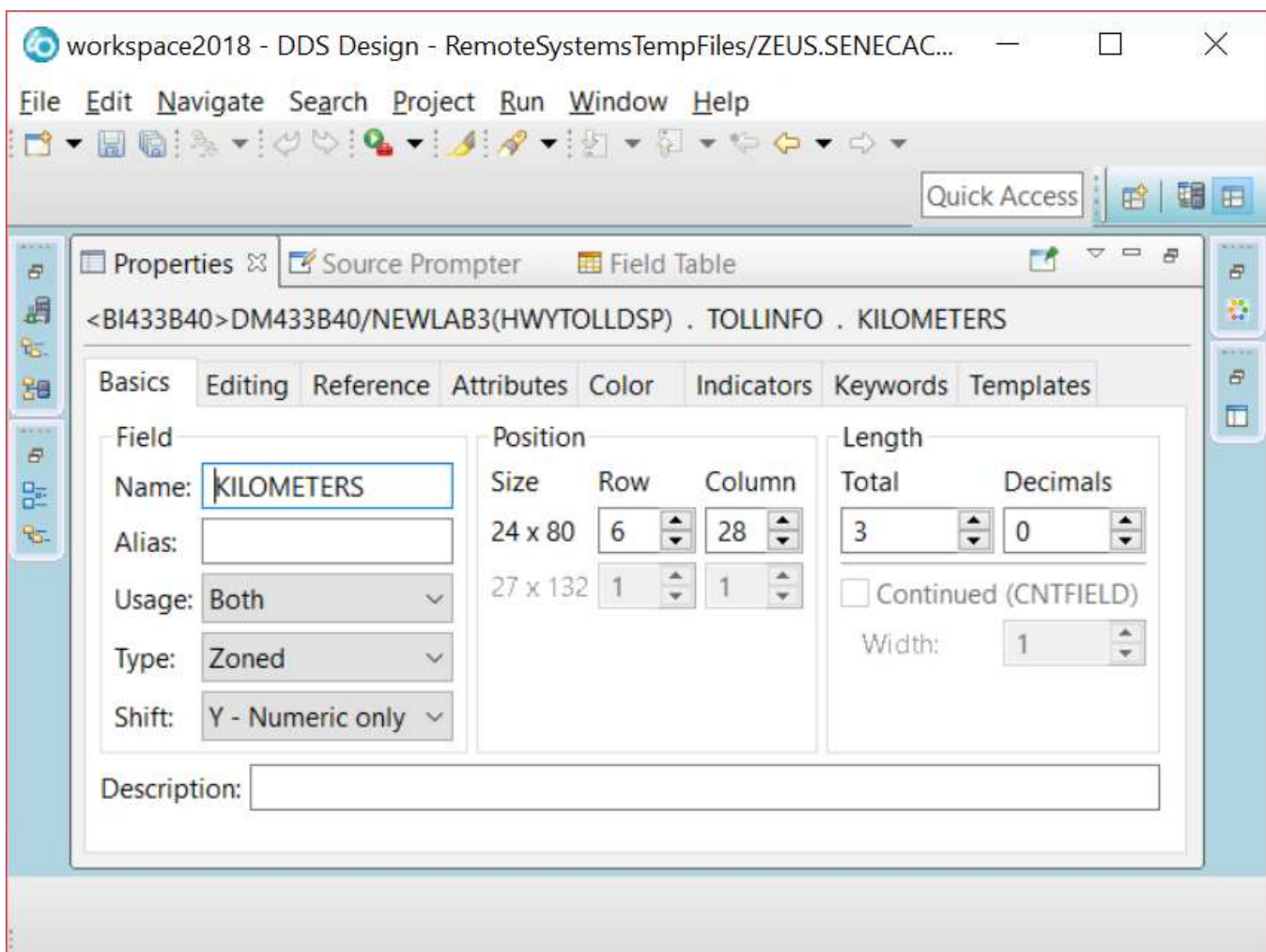
KILOMETERS

We want to make it a **3 digit numeric** field with the field name of **TEST1**. In the Properties view change the type of field from Character to **Zoned**.

What appears on your work screen with the following?

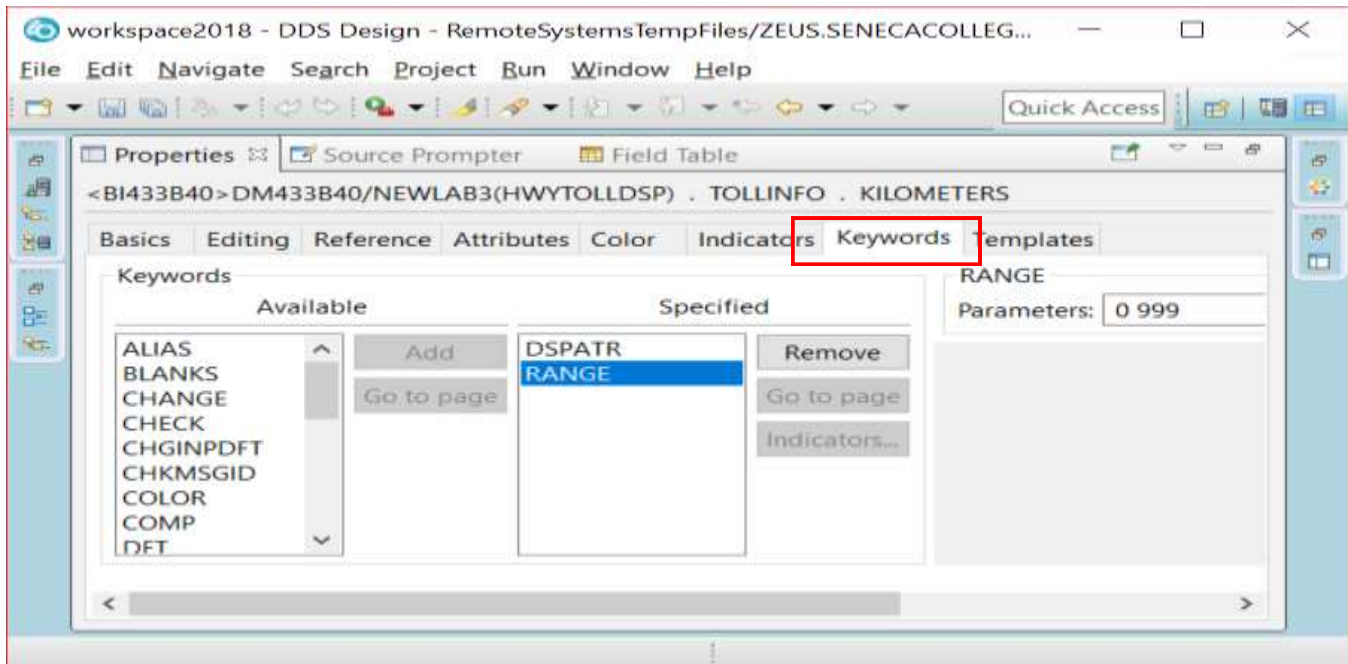
Usage set to INPUT III
Usage set to OUTPUT OOO
Usage set to BOTH BBB

Type and Shift option selections appear below:.

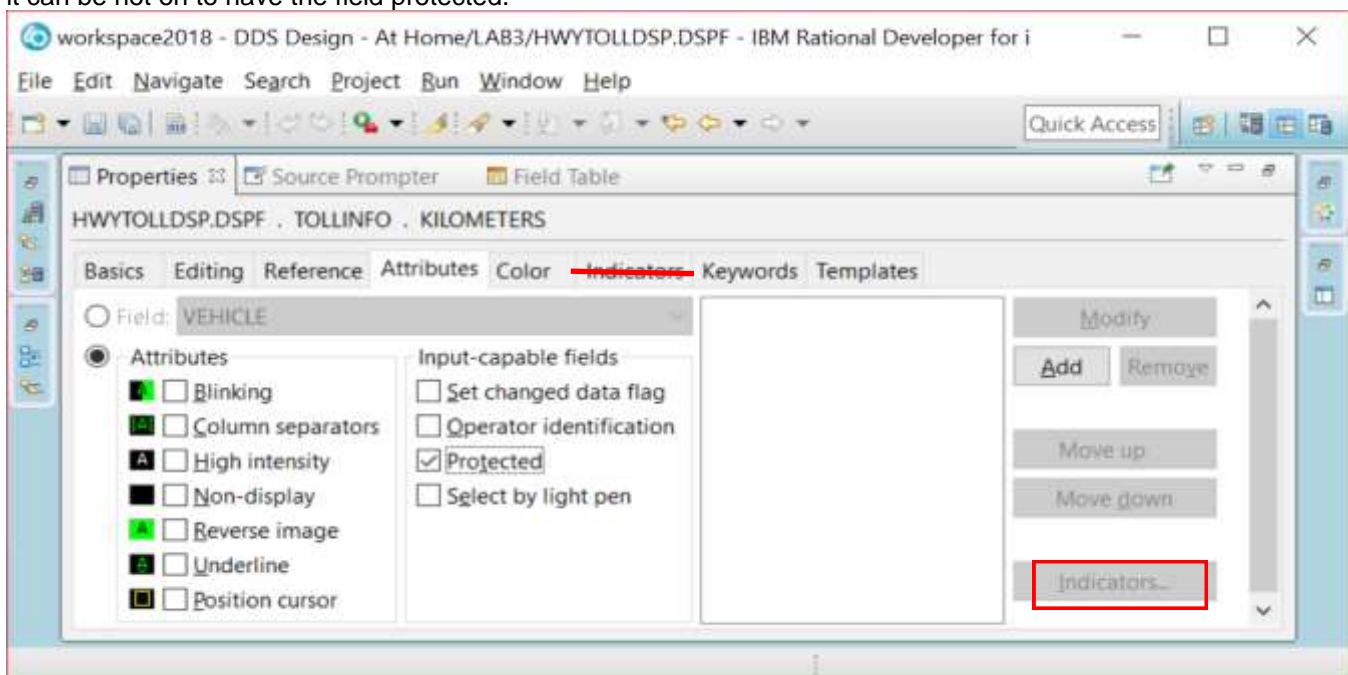


1. Click on the **Keywords** tab and look for the **RANGE** keyword from available keywords on the left hand side. When you click on RANGE, the Add button should become available. Click on this to add RANGE to the specified column box. The Range for a Test mark would be any marks from 0 to 999, Enter this information in the parameters box that should be available while you are working with the specified "RANGE" keyword. (0 999)

A screenshot is provided for this.



- 2.1 The Kilometers field needs to be **protected** when the screen record shows and you do not want to allow input to the field. This can be done by clicking on the **Attributes** tab and then clicking on **Protected**. An **Add** button on the right becomes available so you can add this and this shows in a box on the right. If you click on what became **DSPATR(PR)**
- 2.2 added to the box, you can click on an **Indicators** button on the lower right. You want to enter a **99** in the box as shows on the next page. The yellow box is for the letter N. This indicator can be on to have the field protected or it can be not on to have the field protected.



Add button is lit up, the Indicators button is not lit up until you have added. When used the next screen shows.

Configure Indicators

<input type="checkbox"/>	99	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
or	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
or	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
or	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
or	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
or	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
or	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
or	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
or	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
or	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

OK Cancel Clear

workspace2018 - DDS Design - RemoteSystemsTempFiles/ZEUS.SENECACOLLEG...

File Edit Navigate Search Project Run Window Help

Quick Access

Properties Source Prompter Field Table

<BI433B40>DM433B40/NEWLAB3(HWYTOLLDSP) . TOLLINFO . KILOMETERS

Basics Editing Reference Attributes Color Indicators Keywords Templates

Field: ☐ VEHICLE ☐ 99 DSPATR(PR)

Attributes

- ☐ Blinking
- ☐ Column separators
- ☐ High intensity
- ☐ Non-display
- ☐ Reverse image
- ☐ Underline
- ☐ Position cursor

Input-capable fields

- ☐ Set changed data flag
- ☐ Operator identification
- ☒ Protected
- ☐ Select by light pen

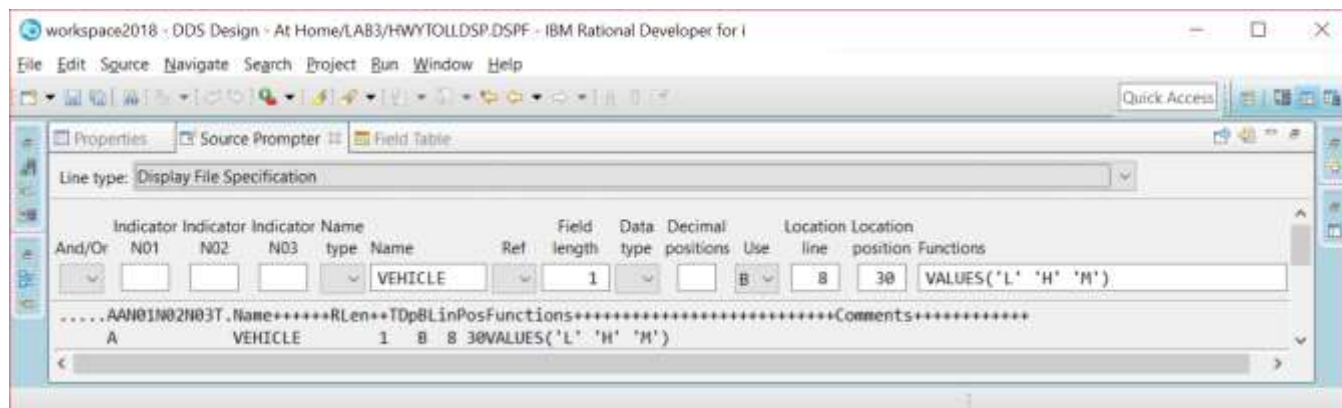
Double click on the Properties tab to get back to the multi view screen

Click on Named Field and place a second field beside a Type of Vehicle: text constant. There are **three tabs** showing midway down your screen. **Design** **Source** and **Preview**. Click on the Source tab and **press F4**.

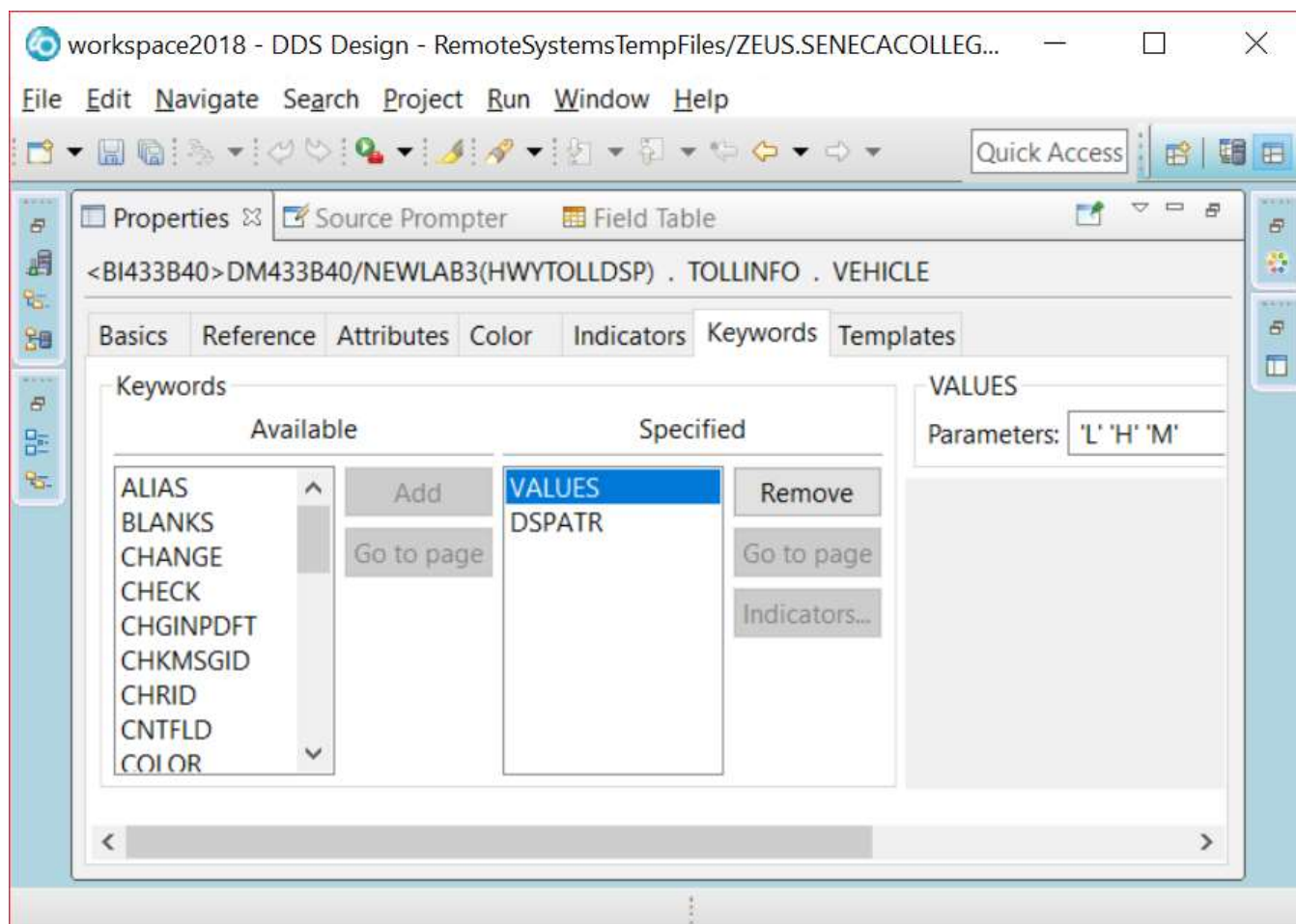
Double click on **the Source Prompter** tab that appears at the bottom view on your screen. On my screen the tabs show up as Remote System Details, Tasks, Object Table, Commands Log, Error List and Source Prompter.
(To add view: Window>Show View>...)

You want to enter the following code.

The one character **Vehicle** field should be limited to the entries of **L, H or M**.

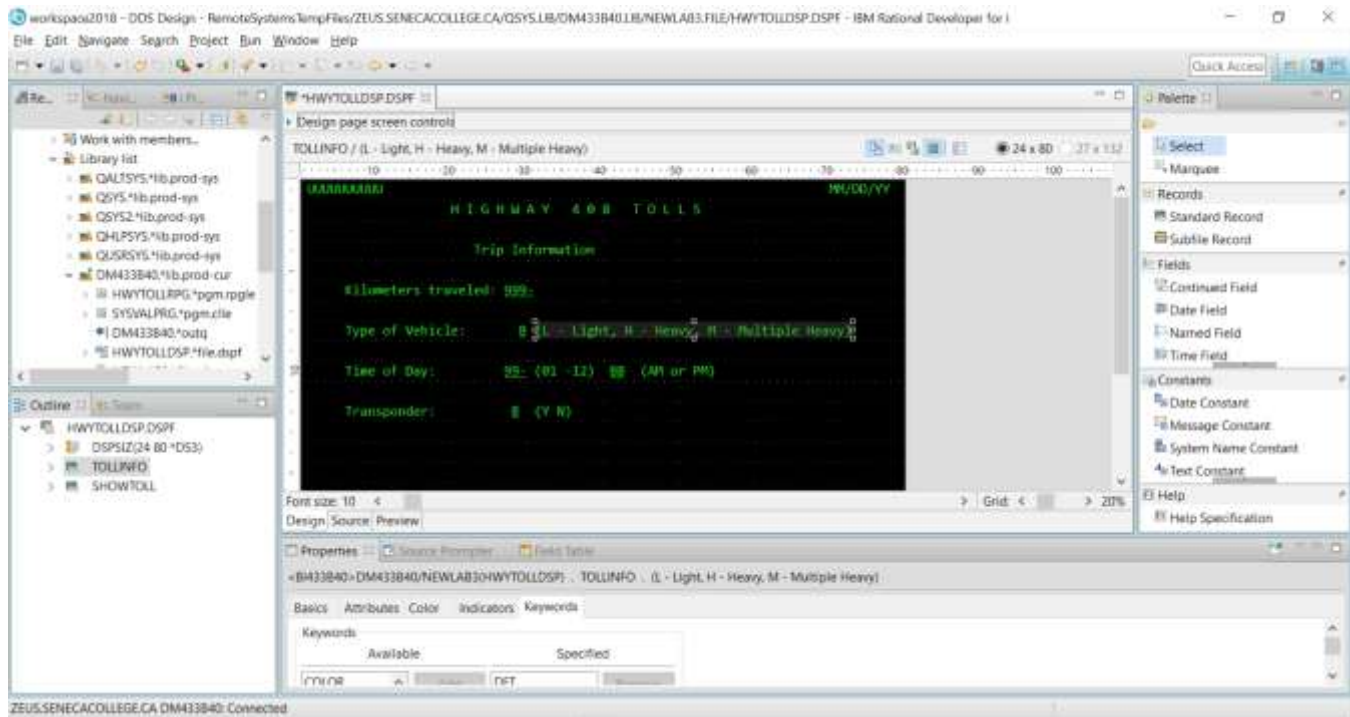


If you do not use the source prompter directly, you can **set the VALUES the same way you set RANGE** for the **VEHICLE** field.



Enter the text and appropriate fields for the “Time of Day” **HOUR**, “(AM or PM)” **AMPM** and “**Transponder**” **TRANSPNDR**. You can also set your properties in the Properties box or use the Source Prompter to enter your code.

You can use the screenshots of the running program as a guide or the finished screen in Screen Designer is shown below. Remember to click on the Design tab to get back to the GUI interface.



DDS code has been generated for you. You can view and change this code. You have to be careful when changing it that your syntax is correct. In order to see the code, click on the **Source** tab. This is found just below your work screen. There are three tabs. **Design** (the one that provides a GUI for creating an interactive screen) **Source** (that allows you to view and change DDS code) and **Preview** (Discussed Later).

workspace2018 - DDS Design - RemoteSystemsTempFiles\ZEUS.SENECACOLLEGE.CA\QSYS.LIB\DM433B40.LIB/...

File Edit Source Compile(G) Navigate Search Project Run Window Help

Quick Access

*HWYTOLLDSP.DSPF

Line	Column	Replace	2 changes	
000900	A	...	AAN01N02N03..	Name+++++RLen++TDpBLinPosFunctions+++++Comments+
001000	A	99		
001001	A			
001100	A			
001200	A			
001300	A	99		
001400	A			
001500	A			
001600	A			
001700	A			
001800	A	99		
001900	A			
002000	A			
002100	A	99		
002200	A			
002300	A			
002400	A			
002500	A	99		
002600	A			
002700	A			
002800	A			
002900	A			
003000	A			

Design Source Preview

ZEUS.SENECACOLLEGE.CA DM433B40: Connected

```

A                                     DSPSIZ(24 80 *DS3)
A
A      R TOLLINFO
A
A      1 2USER
A      1 71DATE
A      EDTCDE(Y)
A      2 21'H I G H W A Y   4 0 8   T O L L S'
A      4 24'Trip Information'
A      6 7'Kilometers traveled:'
A      KILOMETERS 3 0B 6 28
A 99      DSPATR(PR)
A      RANGE(0 999)
A      8 7'Type of Vehicle:'
A      VEHICLE 1 B 8 30VALUES('L' 'H' 'M')
A 99      DSPATR(PR)
A      8 32'(L - Light, H - Heavy, M -
A      Multiple Heavy)'
A      10 7'Time of Day:'
A      HOUR 2 0B 10 28RANGE(1 12)
A 99      DSPATR(PR)
A      10 32'(01 -12)'
A      AMPM 2 B 10 42VALUES('AM' 'PM')
A 99      DSPATR(PR)
A      10 46'(AM or PM)'
A      12 7'Transponder:'

```

```

A          TRANSPNDR      1A  B 12 29
A  99          DSPATR(PR)
A          VALUES('Y' 'N')
A          12 32'(Y N)'
A          R SHOWTOLL
A          CF03(03)
A          OVERLAY
A          13 23'Applicable Toll Charges'
A          15 11'Toll Rate:'
A          TOLLRATE      5Y 40 15 36EDTCDE(1 $)
A          17 9'Toll Charge:'
A          TOLLCHARGE    5Y 20 17 36EDTCDE(1 $)
A          23 3'F3 - Exit'
A          19 10'No Transponder Charge:'
A          NOTRANSF      4 20 19 38EDTCDE(1)
A          21 11'T o t a l   C h a r g e:'
A          TOTALDUE      5 20 21 37EDTWRD(' . ')

```

The length of the ^SKILOMETER field shows as "3 0" which means a length of three and 0 decimal positions for this field. You can see at the top some help for the proper columns to make these sorts of entries. It is easier to use the properties box or the source prompter when entering this information. It is also possible to make direct entries as you get used to the code. Change the length of the KILOMETERS field from three to four and then click on the Design tab to see if this change is shown.

Go back to the Source tab and change it back to a length of three.

We have to be careful here. You could change the "RANGE" keyword to RBNGE" and introduce an error to your code and possibly confuse the Design Tab's interpretation of the code.

According to the DDS code showing on this lab, what line and column(or position) number does my KILOMETERS

appear on? _____ Your code can have this placed on a different line and position.

Go back to the Design tab and place a named field somewhere beside the KILOMETERS field. Check this out in the code by clicking on the Source tab.

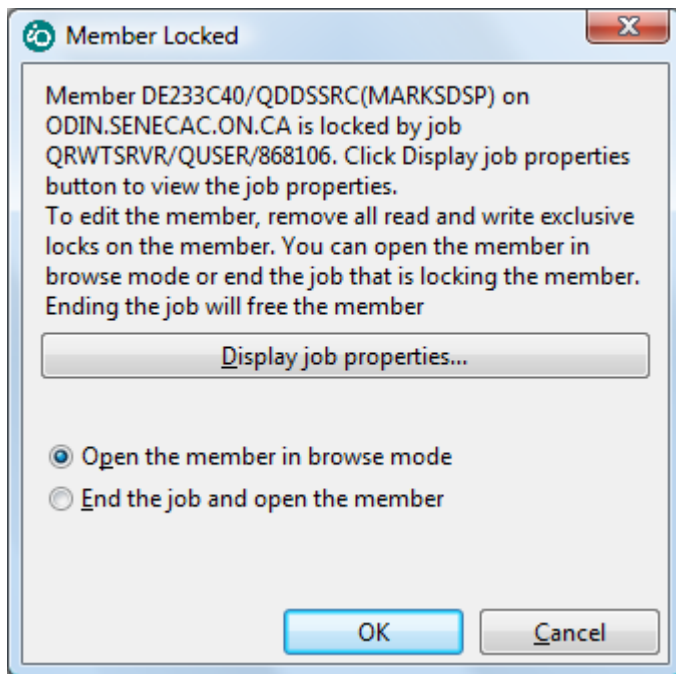
Notice that this new field appears seemingly out of order in the code. It appears at the bottom of the generated code. If you take a look at the line and position number it should show as the same line number you had for your KILOMETERS field.

If this bothers you, you could type an "M" where the line numbers for your code shows (Move). For me that would involve changing 001404 to M01404.

Then put your cursor on the line number for the KILOMETERS field and type an "A" (After). Try it.

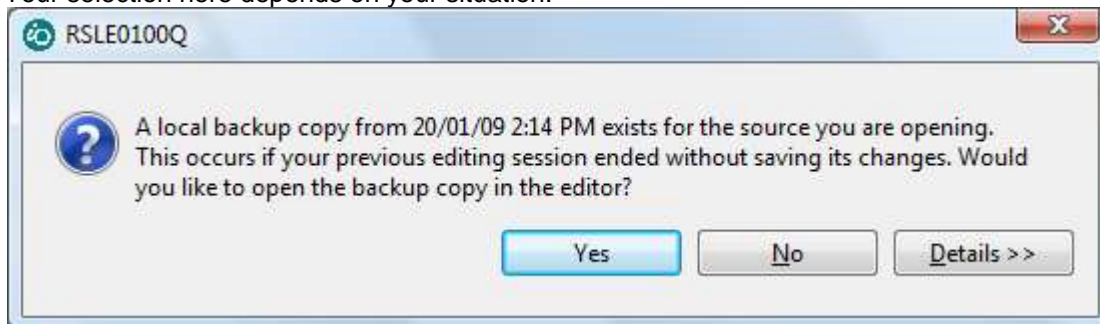
Click on File at the top right hand corner of your screen and select Save. This is a good periodic practice to recover from a crash or other problem.

If you do have a crash, you may restart and get the following message.



You can change the action to End the Job and open the member.

Your selection here depends on your situation.



If you had messed up the current session, you don't want the local backup. If you have done a lot of work and not messed up the current session then you do want the local backup.

You are seeing this error dialogue because while doing this lab, my internet provider caused the connection to drop causing me to have to sign in again. This is less likely to happen to you at school, but may happen at home.

You are ready to provide the second screen record that will overlay the first screen record.

Click on Standard Record in the palette and drop it onto your work area. You should get a RECORD1 with a blank work area. You can toggle between TOLLINFO and RECORD1 by clicking on the appropriate name at the top box in the screen. Try it. Change RECORD1 to the record name SHOWTOLL

Making sure all your entries are below the area for TOLLINFO (in order for OVERLAY to work properly)

Enter:

Constants Field

Toll Rate: a five digit zoned decimal with four decimal positions output only field called **TOLLRATE**
Toll Charge: a five digit zoned decimal with two decimal positions output only field called **TOLLCHARGE**
No Transponder: Charge: a four digit zoned decimal with two decimal positions output only field called **NOTRANSP**

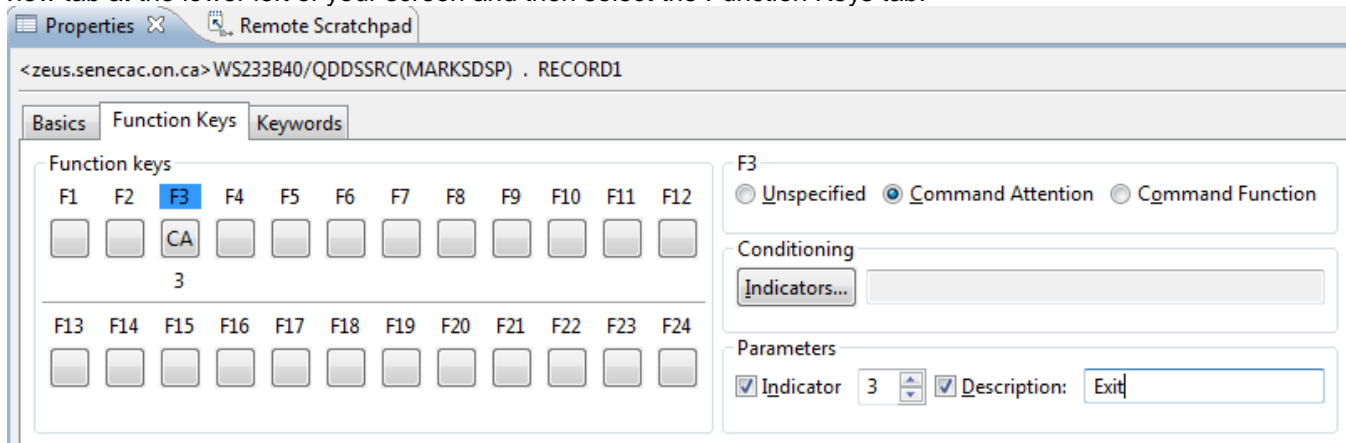
F3- Exit

Specifying Function Keys

We specified a constant which told the user to use F3 to exit the program. Constants are only text, they do not do anything! So, to make the F3 function key available to the user...

SHOWTOLL

Making sure **your focus is not on** a constant or a field in the **RECORD2** work area, double click on the properties view tab at the lower left of your screen and then select the Function Keys tab.



Click on **F3** and on the right hand side select the **Command Attention** radio button.

Under Parameters click on Indicator and beside that change the number to show as a three.

Click on Description and type in EXIT.

Let's review what you have done:

By clicking on F3 **you have enabled the F3 key** for SHOWTOLL. That means when the user is looking at SHOWTOLL they can press F3. They can not press F4 or F5 because those keys have not been enabled. The Enter key is by default enabled for the screen records. Now, the user can press F3 or Enter. Our program will exit when F3 is pressed and reshows TOLLINFO if Enter is pressed.

Command Attention means that no data is passed back to the program. If you entered data into an input capable field, that data would not be available to the program. It is possible to allow an Exit from a program when the user presses F3 and take data entered into the input screen fields and have the program do something with that data before it terminates. ~~Since Command Attention does not allow the program access to the input field data, you would probably select Command Function for that application.~~ **If so,**

The program needs to know if F3 or Enter was pressed. You have enabled a response indicator by checking of **Indicator 3** under Parameters. **There are 99 indicators available to you.** They are like switches with an "On" or an "Off" setting. This logical field either has a '1' or a '0' in it.

Indicators can be viewed as pre-defined variables / switch with two states/values: 'ON'/'OFF' or '1'/'0'

The EXIT Description is a comment that is placed in your code.

What was the actual DDS code for all of this? CA03(03 'Exit')
(You already know how to look this up)

Does this function key setting apply to TOLLINFOR and SHOWTOLL or just one of those records.
apply to SHOWTOLL record only

Add the **OVERLAY** keyword for SHOWTOLL by using the KEYWORDS tab when setting SHOWTOLL properties.

This will allow SHOWTOLL to display without wiping out TOLLINFO. If any of SHOWTOLL's fields or constants are on the same line as a ~~RECORD1~~ field or constant, TOLLINFO will still be wiped out.

Close your HWYTOLLDSP tab and save your work. **So far, we only have DDS code**. We do not have a Display File Object that can be used by programs.

Right click on you closed Display file member (in the Remote Systems View) and **compile it**.

What is the command used to compile a display file DDS member? CRTDSPF FILE(WSONG/HWYTOLLDSP) SRCFILE(...

You should see feedback that the "Events file contains no messages" under the Error List Tab.

Click on the **Commands Log Tab**. You should see dialogue similar to the following:

```
SBMJOB CMD(CRTDSPF SRCFILE(DT433C40/QDDSSRC) SRCMBR(HWYTOLLDSP) REPLACE(*YES)
OPTION(*EVENTF) FILE(DT433C40/HWYTOLLDSP)) JOBD(*USRPRF)
Job 870076/DT433C40/QDFTJOBD submitted to job queue QBATCH in library QGPL.
```

```
CRTDSPF SRCFILE(DT433C40/QDDSSRC) SRCMBR(HWYTOLLDSP) REPLACE(*YES) OPTION(*EVENTF)
FILE(DT433C40/HWYTOLLDSP)
```

File HWYTOLLDSP created in library DT433C40.

If you don't get this there may be an error introduced into the code when you were working with the Source tab. There also may be trouble with your connection. If the **File HWYTOLLDSP created in library** dialogue does not appear and no syntax errors show, then close up RDi, reopen it and then try the compile.

You **can also compile this code using the green screen**. Sign on to a Client Access session.

Type WRKMBRPDM QDDSSRC at the command line.
(This won't work properly if you didn't call your source physical file QDDSSRC)

Use option 14 beside the HWYTOLLDSP member name.

Instead of typing WRKSPLF at the command line to view your listing, type in SP where you had typed in "14".

Use option 5 and scroll up and down your listing or just type in "B" at the top to get to the bottom of your listing and look for the **File HWYTOLLDSP created in library** message.

We are ready to code our program. Here is some help with entering the code.

Although a lot of help is provided, this help assumes that you made it to class and saw your instructor enter the code. If you missed that class, you should be able to get help from a lab aid or an instructor if you are stuck on any step. If you missed both the class and the lab, you should see an BCI433 tutor. in test centre.

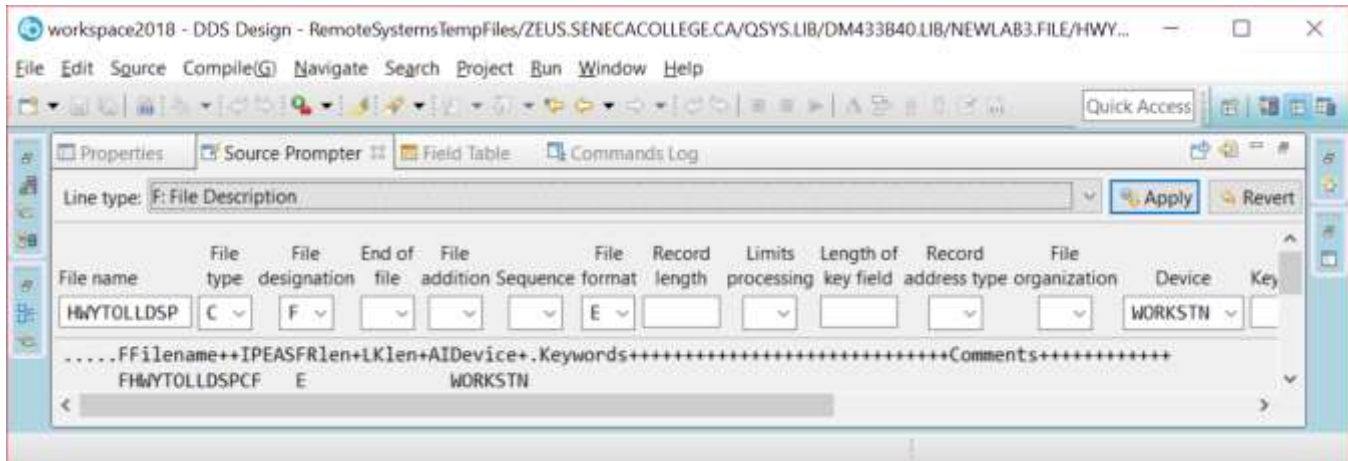
Add a new member called **HWYTOLLRPG** with an **RPGLE member type** to your **QRPGLESRC source physical file**. It will contain your program code for an RPGLE program that uses the display file records you have just created.

Press **F4** to get access to the source prompter and select the F:File Description Line Type.

What file name are we processing? HWYTOLLDSP Enter that in the appropriate prompter box.

Also provide the following:

		Means
<u>File Type:</u>	C	combined or input/output
<u>File Designation:</u>	F	full procedural (you will provide all the logic for the program)
<u>File Format:</u>	E	Externally described file Fields like KILOMETERS will have their definitions brought in automatically from the Display file code. You don't have to define the fields in your program
<u>Device</u>	WORKSTN	A workstation file where input/output is through a display.



Apply this and double click on the HWYTOLLRPG tab.

We are going to enter the rest of our code as Free Format RPG. To do this, we need to enter /FREE on the next line.

The slash must go in column 7. This can be determined by looking at the top area underneath the HWYTOLLRPG tab. As you move your cursor, feedback should be provided about which column number you are in.

The code you need to enter for the **main routine** will be supplied for you. **This version will not be perfect** and may require a few adjustments in a later lab.

It is important to remember that **each program statement needs to end with a semicolon**. If you don't do this, a syntax error may pop up when you go to the next line. The compiler will also reject an RPGLE statement that does not end with a " ; "

If a syntax error pops up it may not always appear immediately below the line that is missing the semicolon. Figure out where you need to make the adjustment and then press CTRL + F5 to get rid of the message that shows up in the LPEX editor.

Common PDM and LPEX editor commands:

A line can be deleted by typing a **D** in the first position of the line number and pressing enter.

A line can also be copied by typing a **C** at the line and a **B** or **A** at the line it is to be copied before or after

Multiple lines can be deleted by typing **DD** at the start of the block and **DD** at the end of the block

CC and **MM** also work with a block of lines

There are more modern ways of editing available with the LPEX editor.

Here is the code you need to enter.

```
/FREE
    EXFMT  TOLLINFO;
        DOW  NOT(*IN03);
            EXSR  SetTollRate;
            *IN99  =      *ON;
            WRITE TOLLInof;
            EXFMT ShowTOLL;
            *IN99  =      *OFF;
            IF *IN03 = *OFF;
                EXSR  CLEAR;
                EXFMT TOLLInfo;
            ENDIF;
        ENDDO;
    *INLR = *ON;
    RETURN;
```

EXFMT is a Read/Write operation

A write of the screen record to the display station and a pause while the user reads it. The user may fill in some fields if there are input capable fields on the screen. At minimum the user will press the return key. A read back to the program occurs and the next program statement is executed.

DOW / ENDDO is for looping. The test is done at the start of the loop and if the test is not passed, none of the loop statements are executed.

NOT(*IN03) is a test of the response indicator 03 which was specified when developing HWYTOLLDSP.

Where was that specified? F3 in record TOLLInfo

You could have also said **DOW (*IN03 = '0')** or **DOW (*IN03 = *OFF)**

EXSR SETTOLLRATE is sending control down to a subroutine called SETTOLLRATE. This subroutine needs to appear after the RETURN statement and will determine the contents of SHOWTOLL output only fields.

Here is some sample code to use for this subroutine:

```
BEGSR SetTOLLRate;
    TOLLRATE   =      .9999;
    TOLLCHARGE =      99;
    NOTRANSP   =      9.99;
    TOTALDUE   =      999.99;
ENDSR;
```

Notice that all those fields have not been determined properly in the code above. You can demonstrate your program with this stub code for this lab and the next lab will include the proper demonstration of tolls according to a chart of values.

Compile HWYTOLLRPG.

Fix any errors

Switch to your Client Access session and run HWYTOLLRPG. This is done by typing

CALL HWY... at the command line.

Enter values for all the input fields and press enter.

When your results screen shows, can you still type data into the top screen fields? _____

Make sure you can exit your program by pressing the F3 key. A previous lab showed what do to if you can't stop your program.

What do you need to do to stop this program if it won't end?

An instructor's sample of this program is available to check yours against. It includes a feature where pressing the F4 key identifies it as a sample program and you will not duplicate this feature. The instructor's version of this program correctly calculates the tolls. You will add this feature for LAB 3B.

To run it **change your current library to BCI433LIB** and then invoke the program:

CHGCURLIB BCI433LIB
CALL HWYTOLLRPG

After you have finished, you should reset your current library back to your own library.

What happens if you do not change the current library when running the instructor's version of this program?

It will depend on the existence of HWYTOLLDSP in your library.

Try running this program with your library as the current library after you have successfully created your version of HWYTOLLDSP.

What happened?

Lab 3 Summary

A display file is coded with DDS code in a member that has a DSPF type.

Once this DDS code has been successfully compiled a Display File object is produced that can be used by different programming languages that want to take advantage of the interactive screens allowing data entry and data display.

The easiest way to define how the display file screens or records will look is to use the Screen Designer GUI in RDi.

The GUI allows you to define field names, types, usage, editing, validations and optional indicators that allow viewing of the field.

You would use the GUI to provide this information and the appropriate line is entered as DDS code.

Here is a sample line with explanations

This is how it actually appears

Ind	Field Name	LT	DU	P	Functions
A 90	TEST1	3S	0B	5 32	<u>RANGE(0 100)</u>

Some separation is shown here to better highlight concepts – the line would not be entered this way.

Ind	Field Name	L	T	D	U	P	Functions
A 90	TEST1	3		S	0	B	5 32 RANGE(0 100)

Indicator 90 has to be on in order to view this field called TEST1 on the screen. If indicator 90 is off, the TEST1 field will not be visible when the screen record is displayed. For our application it would not be a good idea to use an indicator with the TEST1 field.

L shows a field length of three. So at this point XXX could be entered

T or Type shows a field type of zoned decimal and D or Decimal shows 0 decimal positions. So now we cannot put X's in the field. 999 would work.

U or Usage indicates if this field is for input only (I) , output only (O) or both input and output (B).

P or position on the screen shows that this field would reside on line 5 at column 32.

The RANGE function allows us to validate the field before it is passed to a program. The user can enter 0 to 100. They cannot enter a negative number or 101.

Here is some more partial DDS code with explanations

```

A          R RECORD2
A
A          CA03 (03 'EXIT')
A          OVERLAY
A          11 18'Tests:'
A          13 18'Final Mark:'
A          15 18'Final Grade:'
A          22 3'F3=Exit'
A          TESTOVRALL      3Y 00 11 32EDTCDE(1)
A          NUMGRADE        3Y 00 13 32EDTCDE(1)
A          LETGRADE        2A 0 15 33

```

This screen record is called RECORD2 and has some text and fields that are displayed.

At the record level we see CA03 – that enables the pressing of function key F3 when the screen record is displayed

(03 'Exit') indicates when F3 is pressed indicator 03 will be turned on and the comment is 'Exit'

OVERLAY is used to allow RECORD2 to overlay a screen record that is already being displayed. This only works if the first screen record does not use lines 11 – 25. You can see RECORD2 is using lines 11 to 22.

EDTCODE(1) can be used to make a numeric field show with commas, a decimal point and suppressed leading zeros

So 090 would show as 90. If the field was larger 0293334^33 would show as 293,334.33. The decimal point is not stored in the field but can be shown with proper editing. We used the “^” symbol to show where the placeholder for a decimal point is.

RPGLE

A Display file is declared in an RPGLE program with a file specification form

```
FHWYTOLLDSP CF E      WORKSTN
```

FileName	Type	Designation	Format	Device
FHWYTOLLDSP	C	F	E	WORKSTN

Type can be (C)ombined for interactive screens, other options are (I)ntput, (O)utput and (U)pdate

Designation can be (F)ull Procedural – that is the only one we will use this semester

Format can be (E)xternally Described – indicating that the program can bring in field names, types and sizes from the actual file. The programmer does not have to define these fields described in the file.

EXFMT – write a screen record from a display file, pause and when the user presses enter read back what has been inputted into the screen record fields. Sometimes there are no fields and the read back just acknowledges that the screen has been viewed and the user is letting the rest of the program proceed.

DOW - a loop where the test on entering the loop or repeating the loop is done at the start of the loop.

ENDDO

DOU – the loop code is executed at least once and the test on repeating the loop is done at the bottom.

WRITE – can be used to have a screen record display. There is no pause and the program continues. This is useful when showing a screen record and then overlaying a second record and then pausing to let the user look at both screen records.

EXSR GETGRADE – control goes from here to a subroutine at the bottom of the program. The named subroutine has a BEGSR and ENDSR to indicate what code is executed and then control goes back up and executes the next line after the EXSR line.

***IN01 - *IN99** – in RPGLE indicators are referred to by an asterisk and a number. There are 99 indicators available for use and their default setting is *OFF or '0'. They can be turned on during the program run and then checked to see if they are on or off to determine a course of action.

***INLR = *ON** – these two statements are how you should end all your RPGLE programs.

RETURN The last record indicator is turned *ON and RETURN is used to return control to the operating system.