BCI433 Lab 3B (updated Winter 2024)

**Writing an interactive RPGLE screen program**

**Lab objectives:**

* **Apply a mathematical formula to solve an application problem**
* **Use Case structure to solve an application problem**
* **Use the Debug option to check a program’s logic**

**Lab Requirements:**

**Show compiler listing for CARPAYMNT (with Lab3B logic including Case structure)**

**Successfully run CARPAYMNT (with Lab3B logic including Case structure)**

Start an RDi session

Start a ‘Green Screen’ (emulator) Session.

**Using** **Rational Developer for i (RDi):**

**Part A**

**Objectives:**

Case structure is a good choice to solve application logic problems. Here is an example of that:

SELECT;

When Day = 1;

DayName = ‘Monday’;

When Day = 2;

DayName = ‘Tuesday’;

When Day = 3;

DayName = ‘Wednesday’;

When Day = 4;

DayName = ‘Thursday’;

……

ENDSL;

If Day is a 1, then the first test would be done, the DayName would be set to ‘Monday’ and control would go to ENDSL. The test for Day = 2 would not be performed. If Day is a 7 then seven tests would be done and a DayName = ‘Sunday’ statement would be executed.

If an 8 or 9 was entered in Day you could include an Other clause in place of a When clause at the bottom of this structure before the ENDSL. **Other** is a catch all.

Other

DayName = ‘Invalid Day Number’

ENDSL

Tests done with the select When statement can include <=, >=, <> (not equal to), AND, OR

In class exercise:

You need to figure out how to convert a number to a grade.

The mid term test is a mark out of 100 and is worth 35% of the final grade. The labs are entered as a mark out of 30 and are 30% of the final mark. The exam is a mark out of 100 and is worth 35%. If a student fails either the final exam or the test, they will fail the course. The final grade is stored in a field called NUMGRADE. After this grade has been determined, it needs to be converted to a letter grade LETGRADE.

The following information is used to do this.

A+ = 90 - 100 A = 80 – 89 B+ = 75 - 79 B = 70 – 74

C+ = 65 - 69 C = 60 – 64 D+ = 55 - 59 D = 50 - 54

F = 0 – 49

Solution:

Processing for Lab3B

You have an interactive RPGLE program running from lab 3a with no logic to determine the amounts. Added to the purchase price are line items like Freight, Road Ready and Rust Protection which produces a subtotal. The HST tax applied to this subtotal is 13%. The buyer can lower his loan amount by including a trade in, qualifying for a rebate and providing a down payment. This money applied lowers the principal owing for the loan.

The amount for rust protection is based on the basic vehicle cost. It is set at 1.8% of the vehicle price. If rust protection is refused it would be 0.

The standard formula applied to a loan amount to determine payments over a term at an interest rate is the following: note the exponent reads as (-number of months)

Table

Description automatically generated

After you have calculated the payment, you can include and calculate for two new fields on your screen.

The TOTALOFALL which is how much was paid over the entire term and TOTALINT which is how much interest you paid over the entire term of the loan.

Apply the information and features below to determine the correct output for completion of lab 3b.

A close-up of a receipt

Description automatically generated

Include an inline case solution to possible adjustments to the interest rate if the term is not 49 to 60 months.

|  |  |
| --- | --- |
| TERM (in months) |  |
| O |  |
| 1-36 | Interest rate adjusted down by .77 |
| 37 – 48 | Interest rate adjusted down by .6 |
| 49 – 60 | No adjustments to interest rate |
| 61 – 72 | Interest rate adjusted up by .65 |
| 73 – 80 | Interest rate adjusted up by 1.3 |
| 81 - ?? | Interest rate set to 99.99 |

Here is an example where an interest rate of 4.5% was input, but it got adjusted due to the default term of 60 months being changed to 80 months.

A screenshot of a credit card

Description automatically generated

**Enter was pressed.**

A screenshot of a car purchase receipt

Description automatically generated

The user has the option to exit the program or enter more vehicle purchase transaction scenarios to see how that impacts the monthly payments.

The best way to check the behaviour of the program is to run the instructor version.

ChgCurLib BCI433LIB

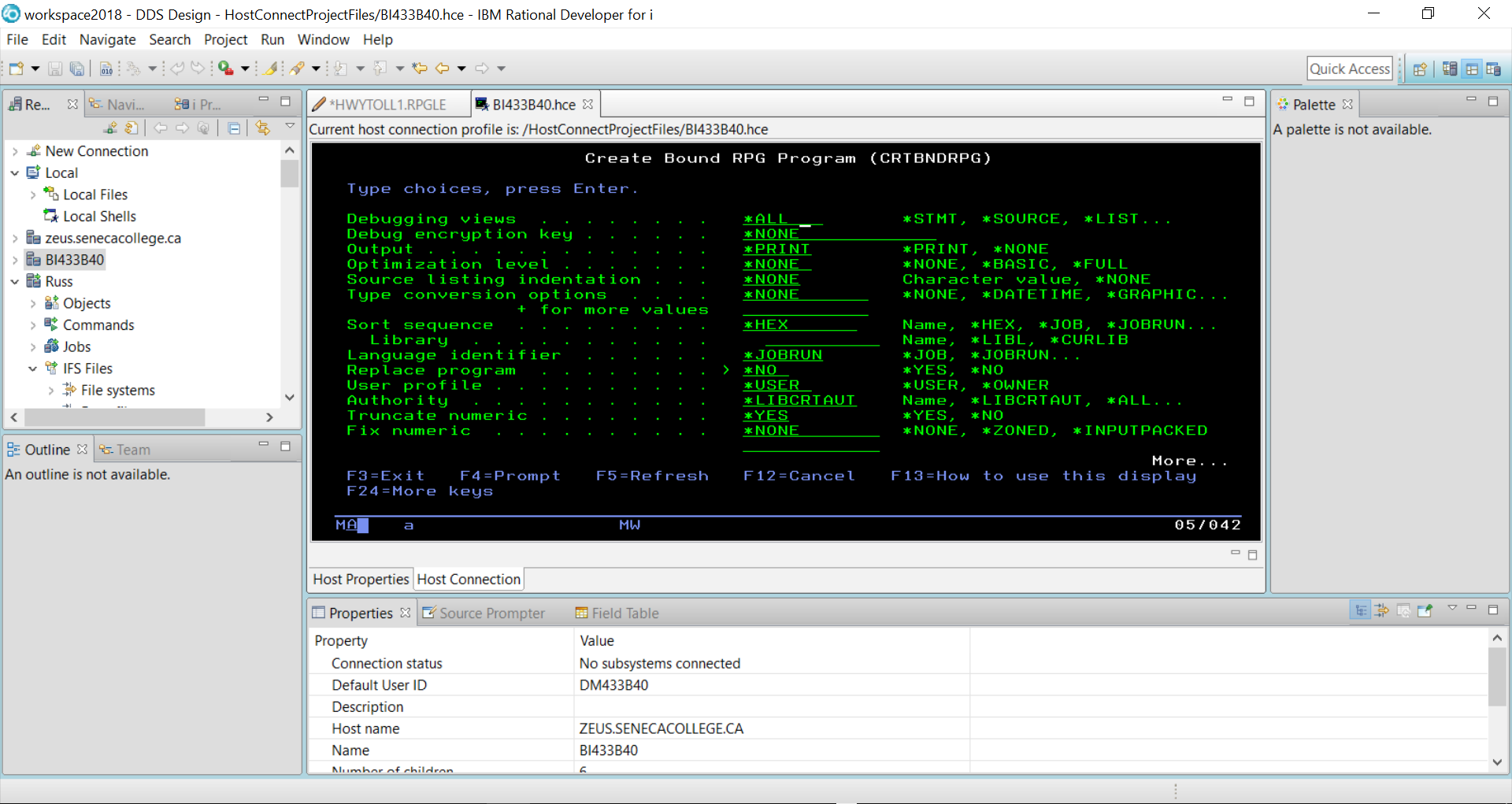
Call CARPAYMNT2

ChgCurLib (Back to your library)

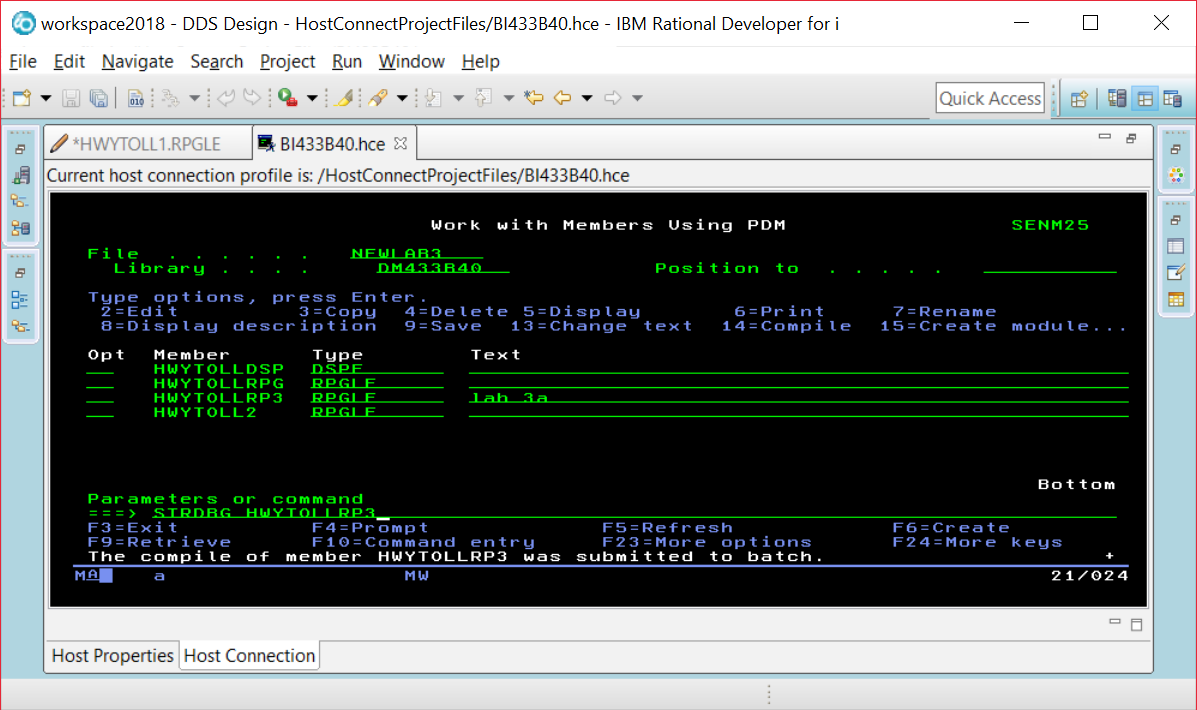
Your program should get similar results. In order to test your program, your professor will try a few scenarios, scan your PDF listing to see if there are any obvious errors. Your listing can be printed by a PC printer and should have utilized the CALL STRJOB command before compilation, so your id and name appear at the bottom of each page.

Using Debug With an Interactive RPGLE Program

Green screen compile requires an option to be changed. Debugging views should be set to all. RDi compile does this by default.

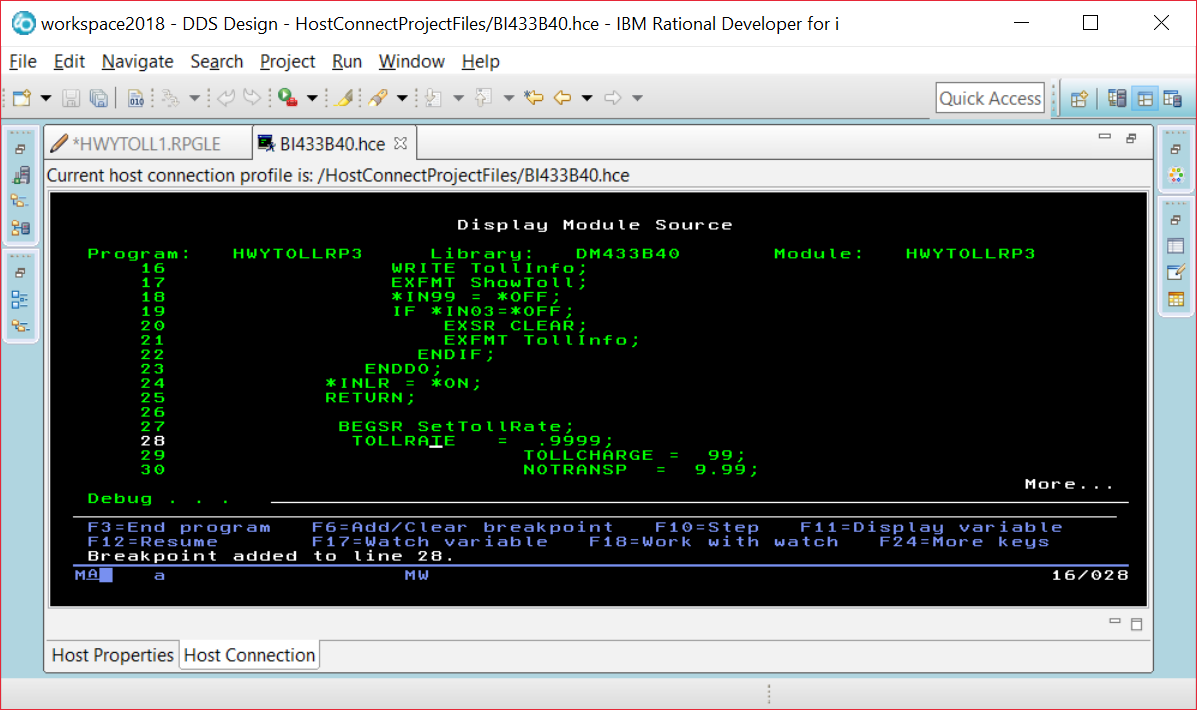


The STRDBG command allows you to set a breakpoint in the program when it is running.

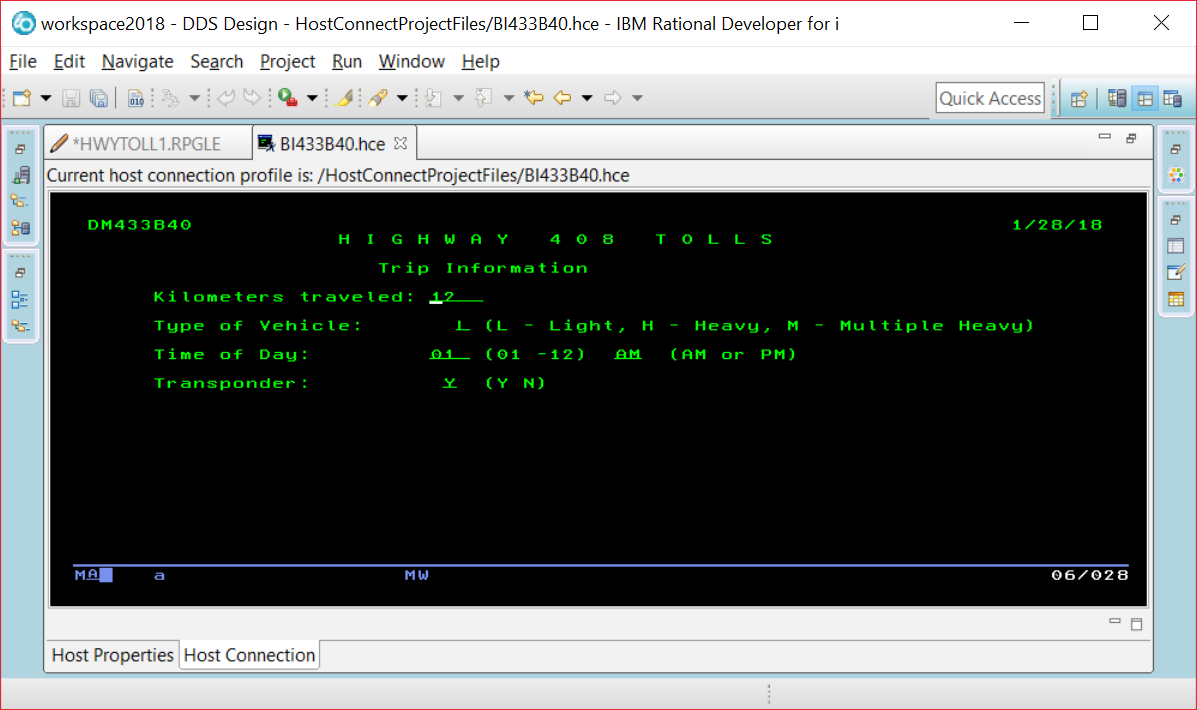


If you are already in DEBUG mode, this command will not work. Just enter the ENDDBG command to exit from DEBUG mode and then enter the STRDBG command with your program name.

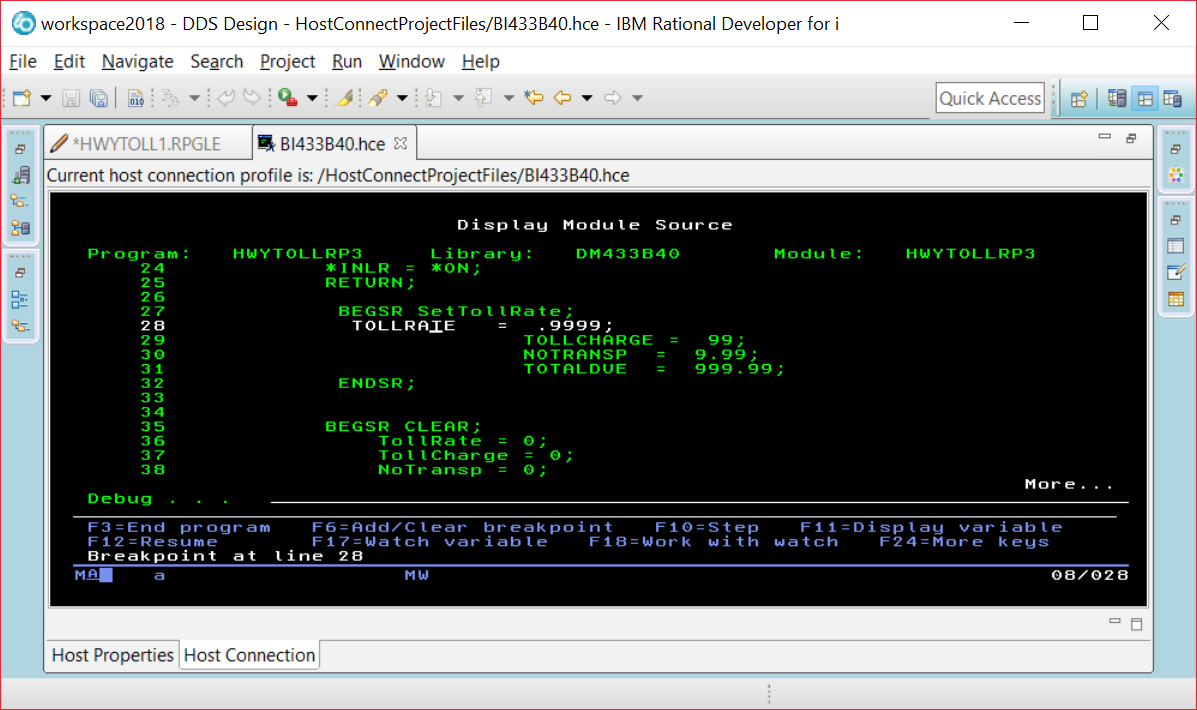
Page down and put your cursor on the line where you want a breakpoint, and press F6.



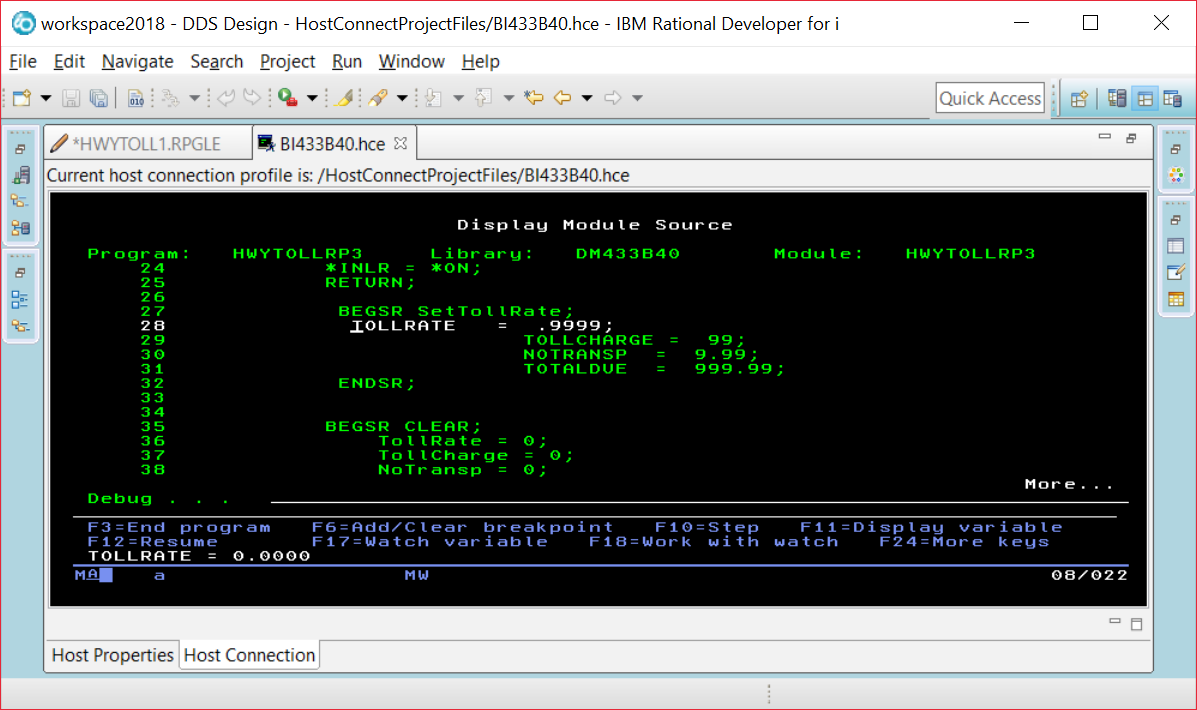
When you run the program you will see screens the program displays before the breakpoint. Our breakpoint was after the EXFMT line, so we see this screen and can enter data into the fields.



The breakpoint is reached and this line has not been executed yet. So TOLLRATE would still be set at 0.

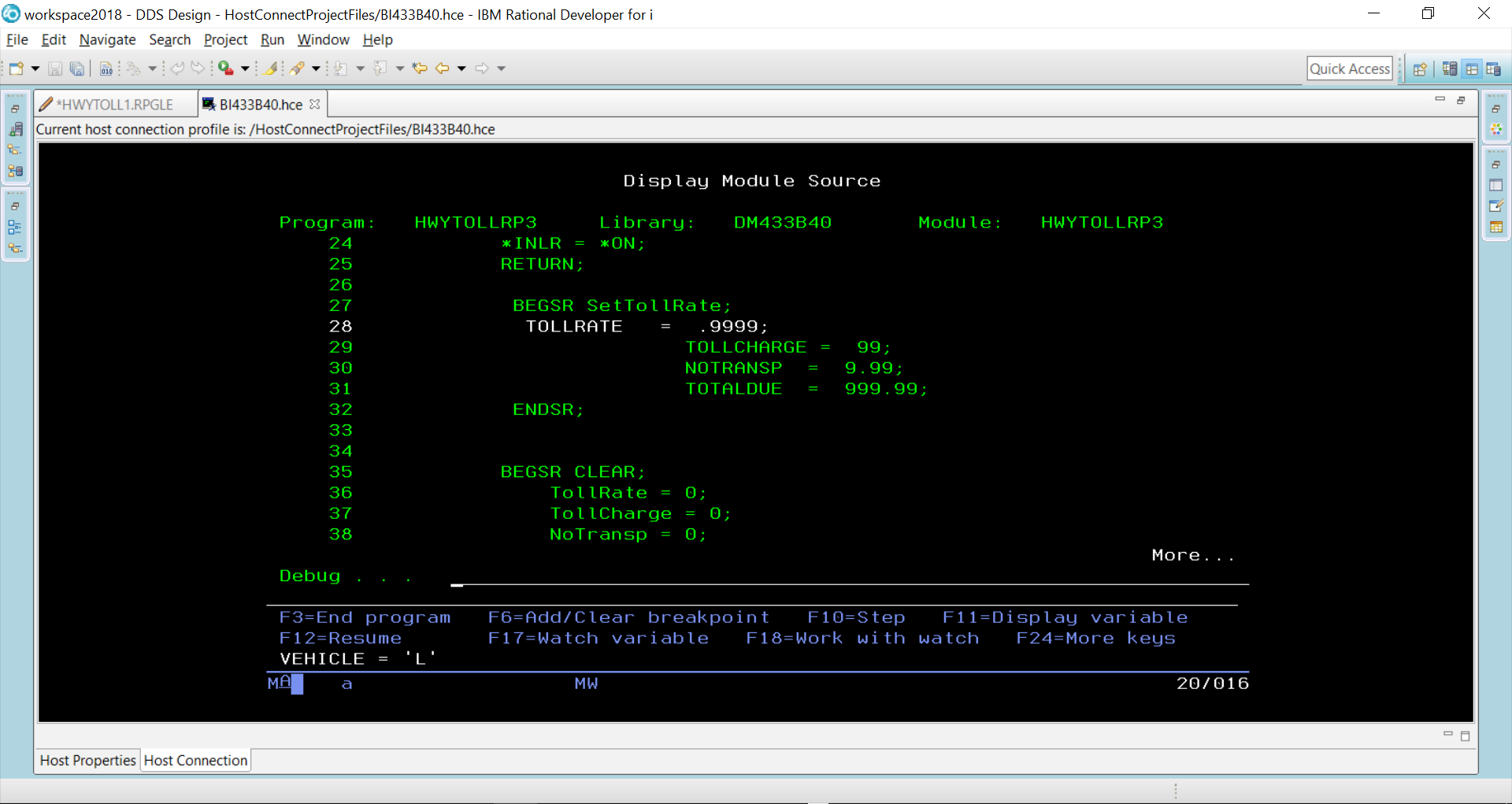


If you put your cursor on the TOLLRATE field, you will see what is currently stored in the field.



Instead of using F11, you run the command EVAL VEHICLE command at the prompt and be able to find out what is in a field that is not currently being displayed in a line of code on your screen.

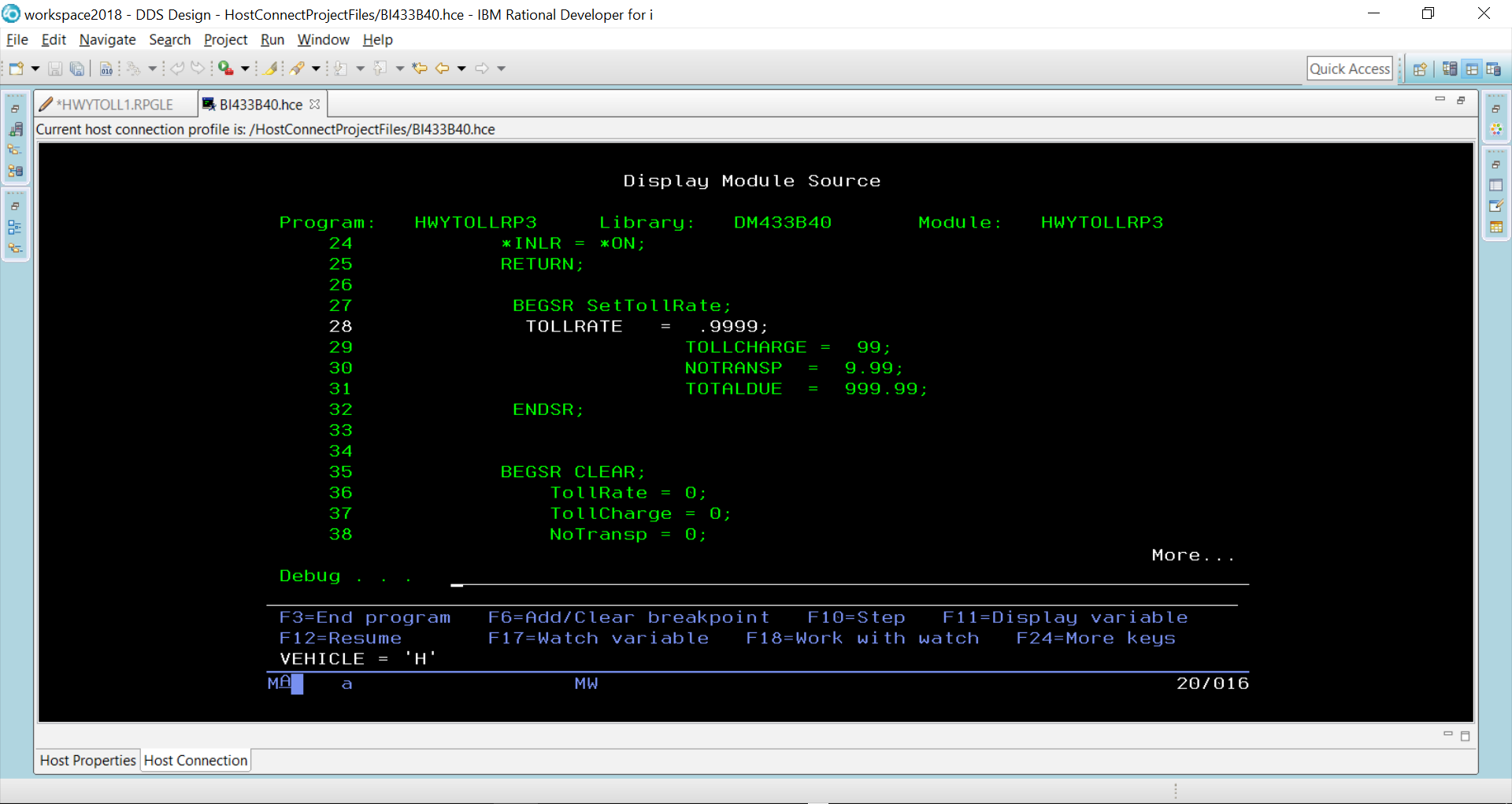
EVAL VEHICLE



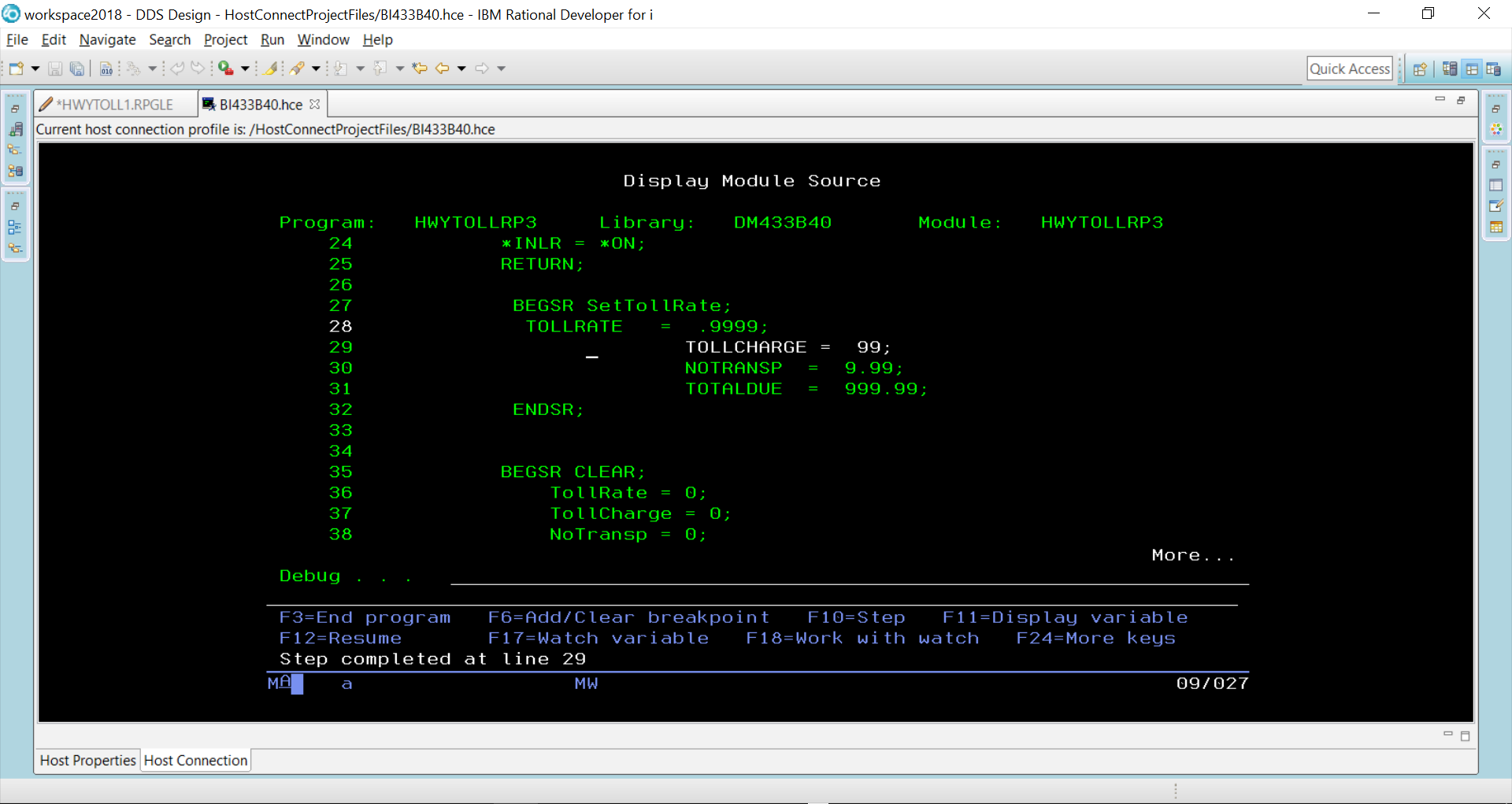
You can alter the contents of a field while your program is running.

EVAL VEHICLE = ‘H’

EVAL VEHICLE



F10 is pressed to step through each line of code



TollRate would now be set at 9999 because line 28 has executed.

When F12 pressed – the rest of the code executes until we get to where the two screens are made available and the program is paused.

