

Make three source members: THETABLE, THESTRUCT, and THEPROGRAM.

Copy the sources below into each member as appropriate.

Source code for table THETABLE:

```
create or replace table THETABLE (  
  A_LONGER_FIELD_NAME for FIELD1 char(10) not null with default  
  ) rcdfmt THETABLER;
```

Source code for table THESTRUCT:

```
create or replace table THESTRUCT (  
  JERRY_LEWIS for FIELD1 decimal(9) not null with default  
  );
```

Source code for THEPROGRAM:

```
**free

ctl-opt text( 'Demonstrate refactor defect' )
  main( THEPROGRAM )
  debug( *constants: *retval ) actgrp( *new )
  option( *srcstmt: *nodebugio: *nounref: *noshowcpy: *noexpdds: *showskp );

dcl-f THETABLE usage(*input);

dcl-proc THEPROGRAM;
  dcl-pi *n;
  end-pi;

  dcl-s tempNumber packed( 9 );

  read THETABLE;

  if FIELD1 <> *blanks;
    THESUBPROCEDURE( tempNumber );
  endif;

on-exit;
  close *all;

end-proc;

dcl-proc THESUBPROCEDURE;
  dcl-pi *n;
  someField like( FIELD1 );
  end-pi;

  dcl-ds THESTRUCT extname( 'THESTRUCT' ) end-ds;

  if FIELD1 > *zero;
    someField = FIELD1;
  endif;

  clear THESTRUCT;

end-proc;
```

Compile both tables into a library. Make sure that library is in your library list.

When you open the program source, you can see RDi builds the outline incorrectly. Maybe that's a clue.

The screenshot displays the IBM Rational Developer for i (RDi) interface. The left pane shows the source code for 'THEPROGRAM.RPGLE'. The right pane shows the 'Outline' view, which is incorrectly generated.

Source Code (THEPROGRAM.RPGLE):

```
Line 23      Column 32      Insert      Browse
...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8...+...9...+...0
000100  **free
000200
000300  ctl-opt text( 'Demonstrate refactor defect' )
000400  main( THEPROGRAM )
000500    debug( *constants: *retval ) actgrp( *new )
000600    option( *srcstmt: *nodebugio: *nounref: *noshowcpy: *noexpdds: *showskp );
000700
000800  dcl-f THETABLE;
000900
001000  dcl-proc THEPROGRAM;
001100    dcl-pi *n;
001200    end-pi;
001300
001400    dcl-s tempNumber packed( 9 );
001500
001600    read THETABLE;
001700
001800    if FIELD1 <> *blanks;
001900      THESUBPROCEDURE( tempNumber );
002000    endif;
002100
002200  on-exit;
002300    close *all;
002400
002500  end-proc;
002600
002700  dcl-proc THESUBPROCEDURE;
002800    dcl-pi *n;
002900    someField like( FIELD1 );
003000    end-pi;
003100
003200    dcl-ds THESTRUCT extname( 'THESTRUCT' ) end-ds;
003300
003400    if FIELD1 > *zero;
003500      someField = FIELD1;
003600    endif;
003700
003800    clear THESTRUCT;
003900
004000  end-proc;
```

Outline View:

- Global Definitions
 - Files
 - THETABLE: DISK EXTDESC('SRQUICKIE/THETABLE')
 - THETABLER
 - FIELD1: CHAR(10)
 - 29
 - 18
 - 34
 - 35
 - 16 - READ
 - Fields
 - FIELD1: CHAR(10)
 - 29
 - 18
 - 34
 - 35
 - Prototypes
 - THEPROGRAM: EXTPROC ('THEPROGRAM')
 - 4
 - 10
 - THESUBPROCEDURE: EXTPROC ('THESUBPROCEDURE')
 - Parameters
 - someField: CHAR(10)
 - 27
 - 19
 - Subprocedures
 - THEPROGRAM
 - Local Definitions
 - Fields
 - tempNumber: PACKED(9: 0)
 - 19
 - 4
 - 10
 - THESUBPROCEDURE
 - Parameters
 - someField: CHAR(10)
 - 35 (M)
 - Local Definitions
 - Data Structures
 - THESTRUCT: EXTNAME(@SRQUICKIE/THESTRUCT: THESTRUCT)
 - FIELD1: PACKED(9: 0)
 - 38 (M)
 - Fields
 - FIELD1: PACKED(9: 0)
 - someField: CHAR(10)
 - 35 (M)
 - 27
 - 19

Now, in THEPROGRAM, highlight the THETABLE file name and use the “Refactor | Rename to use file ALIAS” menu option. Result no longer compiles. I think it is because the scope of the outline data picked up the wrong FIELD1 in THESUBPROCEDURE initially.

The screenshot shows the SAS Studio interface with the code editor and the outline pane. The code editor displays the source code for THEPROGRAM.RPGLE, with line 22 highlighted. The outline pane on the right shows the project structure, including Global Definitions, Files, Fields, Prototypes, and Subprocedures. The THETABLE file is highlighted in the Files section, and the THESUBPROCEDURE is highlighted in the Subprocedures section.

```
Line 22      Column 46      Insert
...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8...+...9...+...0
000100 **free
000200
000300 ctl-opt text( 'Demonstrate refactor defect' )
000400 main( THEPROGRAM )
000500 debug( *constants: *retval ) actgrp( *new )
000600 option( *srcstmt: *nodebugio: *nounref: *noshowcpy: *noexpdds: *showskp );
000700
000800 dcl-f THETABLE usage(*input) ALIAS;
000900
001000 dcl-proc THEPROGRAM;
001100   dcl-pi *n;
001200   end-pi;
001300
001400   dcl-s tempNumber packed( 9 );
001500
001600   read THETABLE;
001700
001800   if A_LONGER_FIELD_NAME <> *blanks;
001900     THESUBPROCEDURE( tempNumber );
002000   endif;
002100
002200 on-exit;
002300   close *all;
002400
002500 end-proc;
002600
002700 dcl-proc THESUBPROCEDURE;
002800   dcl-pi *n;
002900   someField like( A_LONGER_FIELD_NAME );
003000   end-pi;
003100
003200   dcl-ds THESTRUCT extname( 'THESTRUCT' ) end-ds;
003300
003400   if A_LONGER_FIELD_NAME > *zero;
003500     someField = A_LONGER_FIELD_NAME;
003600   endif;
003700
003800   clear THESTRUCT;
003900
004000 end-proc;
```

(THEPROGRAM)

Outline

type filter text

- Control Statements
- Global Definitions
 - Files
 - THETABLE : DISK usage(*input) EXTDESC('@SRQUICKIE/THETABLE') ALIAS
 - THETABLER
 - A_LONGER_FIELD_NAME : CHAR(10)
 - 16 - READ
 - Fields
 - A_LONGER_FIELD_NAME : CHAR(10)
 - 29
 - 18
 - 34
 - 35
 - Prototypes
 - THEPROGRAM : EXTPROC ('THEPROGRAM')
 - 4
 - 10
 - THESUBPROCEDURE : EXTPROC ('THESUBPROCEDURE')
 - Parameters
 - someField : CHAR(10)
 - 27
 - 19
 - Subprocedures
 - THEPROGRAM
 - Local Definitions
 - Fields
 - tempNumber : PACKED(9: 0)
 - 4
 - 10
 - THESUBPROCEDURE
 - Parameters
 - someField : CHAR(10)
 - 35 (M)
 - Local Definitions
 - Data Structures
 - THESTRUCT : EXTNAME(@SRQUICKIE/THESUBPROCEDURE : THESTRUCT)
 - FIELD1 : PACKED(9: 0)
 - 38 (M)
 - Fields
 - FIELD1 : PACKED(9: 0)
 - someField : CHAR(10)
 - 35 (M)