Rocket® Visual COBOL® (formerly a product of Micro Focus or formerly a product of Open Text) documentation that was created prior to Rocket Software's acquisition of certain products from OpenText may contain outdated references to "Micro Focus" or "OpenText", which are trademarks of OpenText. Rocket Software is not affiliated with Micro Focus or OpenText and has rebranded these products as Rocket® products.

Rocket software

Visual COBOL for Eclipse on Windows V10.0

Notices

Copyright

Trademarks

Examples

This information might contain examples of data and reports. The examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

License agreement

Corporate information

Rocket Software, Inc. develops enterprise infrastructure products in four key areas: storage, networks, and compliance; database servers and tools; business information and analytics; and application development, integration, and modernization.

Website: www.rocketsoftware.com

Rocket Global Headquarters

77 4th Avenue, Suite 100

Waltham, MA 02451-1468

USA

To contact Rocket Software by telephone for any reason, including obtaining pre-sales information and technical support, use one of the following telephone numbers.

Country and Toll-free telephone number

- United States: 1-855-577-4323
- Australia: 1-800-823-405
- Belgium: 0800-266-65
- Canada: 1-855-577-4323
- China: 400-120-9242
- France: 08-05-08-05-62
- Germany: 0800-180-0882
- Italy: 800-878-295
- Japan: 0800-170-5464
- Netherlands: 0-800-022-2961
- New Zealand: 0800-003210
- South Africa: 0-800-980-818
- United Kingdom: 0800-520-0439

Contacting Technical Support

The Rocket Community is the primary method of obtaining support. If you have current support and maintenance agreements with Rocket Software, you can access the Rocket Community and report a problem, download an update, or read answers to FAQs. To log in to the Rocket Community or to request a Rocket Community account, go to www.rocketsoftware.com/support. In addition to using the Rocket Community to obtain support, you can use one of the telephone numbers that are listed above or send an email to support@rocketsoftware.com.

Table of contents

Debugging Java calling COBOL and accessing COBOL working-storage (separate projects)

You need to run through the *Java calling COBOL and accessing COBOL working-storage (separate projects)* example before attempting this, as it creates the projects required (**JCallShare** and **CCallShare**) to complete these steps.

The attach to process technique used here can also be applied to the other examples in this section that deal with separate projects.

1. Set two breakpoints in the Java program (`Demo4.java`):
    a. From within **JCallShare**, open `Demo4.java` in the editor.
    b. Set breakpoints on line 9 (the line that reads `System.out.println("--Java--");`) and line 37 (the line that reads `progs.demo4();`).
2. Set a breakpoint in the first COBOL program (`demo4.cbl`):
    a. From within **CCallShare**, open `demo4.cbl` in the editor.
    b. Set a breakpoint on line 15 (the line that reads `display "--In COBOL Program demo4--"`).
3. Click  to the right of  (**Debug**) and select the run configuration that you used to run `Demo4.java` in the previous example.
4. If you are prompted to switch to the Debug perspective, click **Switch**.

    The Debug perspective is displayed, and the debugger runs until it hits the first breakpoint set in `Demo4.java`.

5. Use the usual debug commands to step into, step over, etc... until you reach the next breakpoint (line 37 - `progs.demo4();`).

    > Shortcut: Click  (**Resume**) to jump to the breakpoint.

    At this point, only `Demo4.java` is on the debug stack. We need to also attach `demo4.cbl`, so that we can also step though that code.

6. Attach `demo4.cbl` to the running Java debug process:
    a. Click  to the right of  (**Debug**) and select **Debug Configurations**.

        The **Debug Configurations** dialog box is displayed.

    b. In the left-hand pane, double-click **COBOL Attach to Process**.

        A new configuration is displayed in the right-hand pane.

    c. In the **Name** field, enter a name for the configuration.
    d. Click **Browse** to the right of the **COBOL Project** field, and then select **CCallShare**.
    e. Click **Debug**.

        The **Select Process** dialog box is displayed.

    f. Select the Java process currently being debugged, and then click **OK**.

> Tip: The process is named `javaw.exe`; if you have more than one entry, match the process ID with the one displayed in the debug stack.

The `demo4.cbl` program is added to the stack.

7. Within the debug stack, ensure that the `Demo4.main` entry is selected, and then click (**Resume**) again.

   Debugging flow continues to `demo4.cbl` and stops on the breakpoint set previously.

8. Continue to debug as normal.

This demonstrates how you can debug through both the Java code and the COBOL code.

Related information

Java calling COBOL and accessing COBOL working-storage (separate projects)