

Eclipse

CommitWork Developer Suite CDS

Februar 2010

Jürgen Depping / Hans Fried Kirschbaum
© CommitWork GmbH

CommitWork

GmbH für Informationstechnologie

Info@CommitWork.de

www.CommitWork.de



Softwareentwicklung auf der NonStop, ein steiniger Weg.



Muss nicht sein, es gibt Eclipse-plugins für Nonstop

Einsatz von Eclipse für traditionelle Programmierung

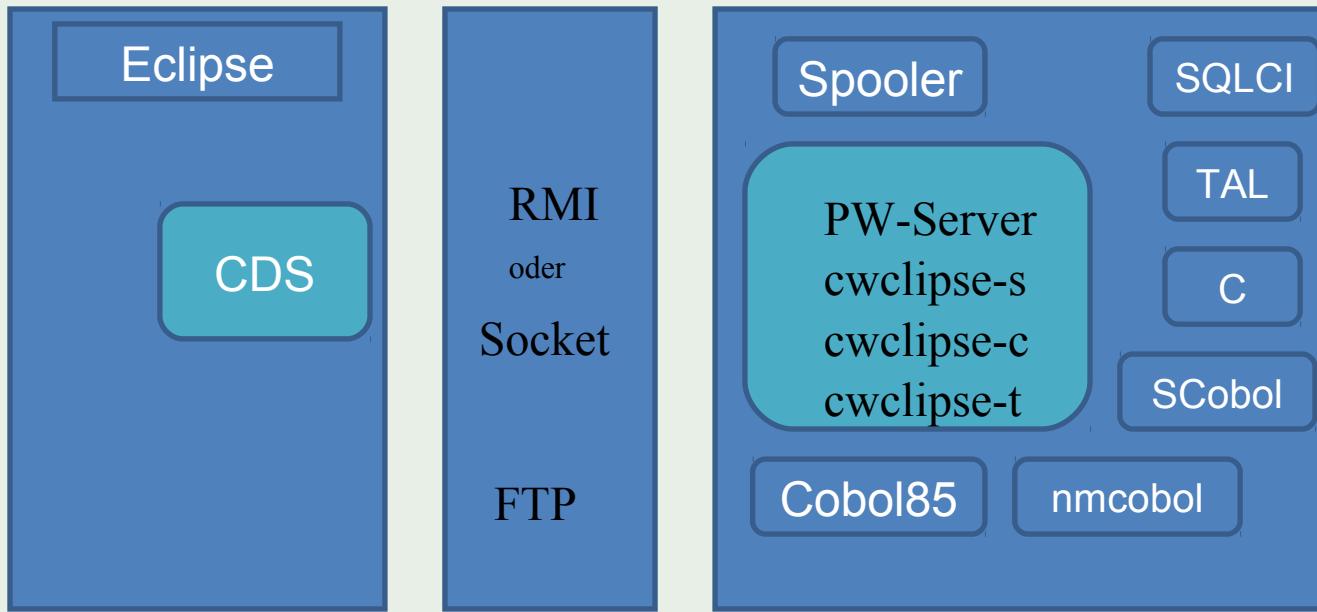
HP: Native-Code Crosscompiler

Vorteil: Entlastung der NonStop
Lokales debugging

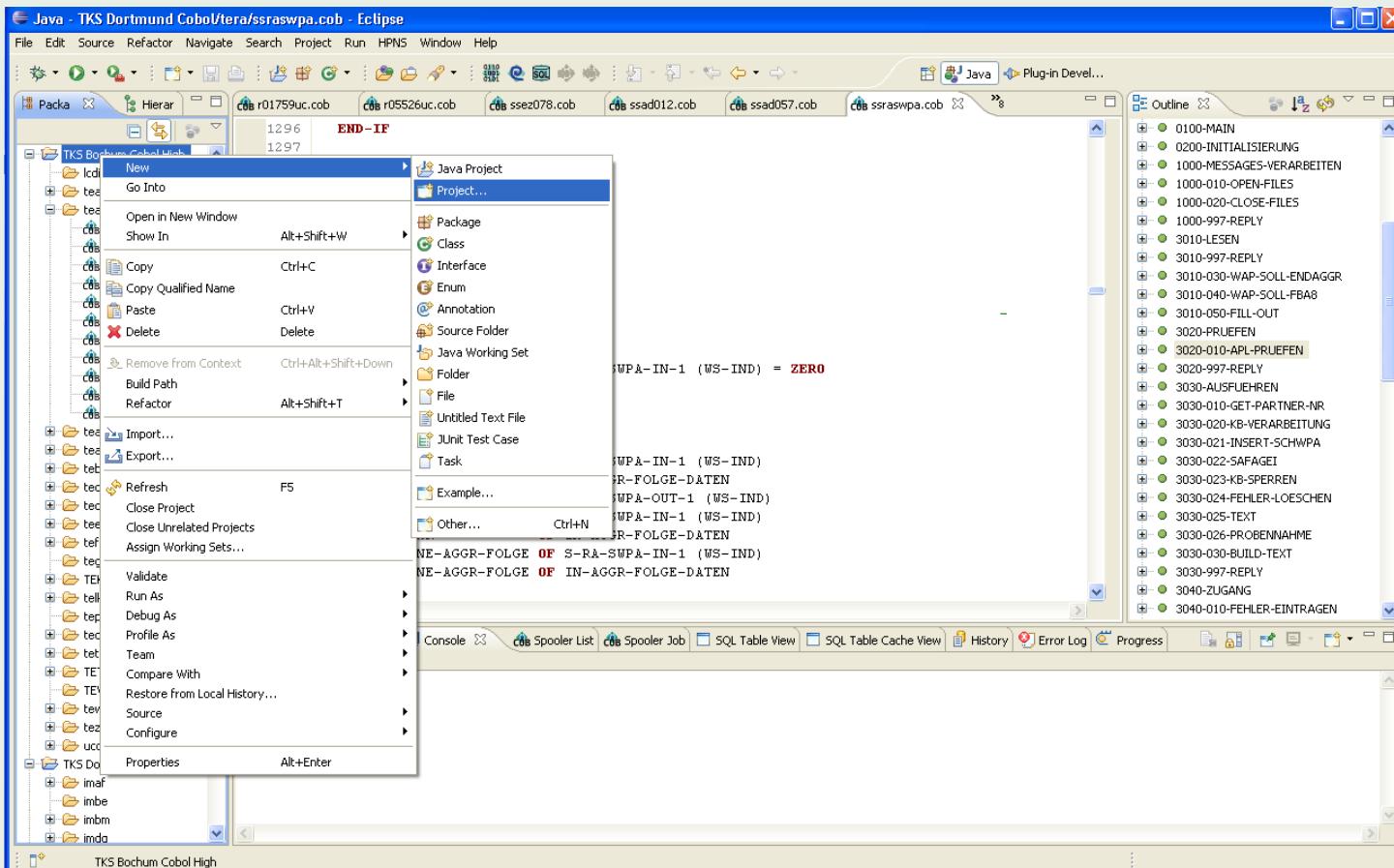
CommitWork: Compile auf der NonStop

Vorteil: Benutzung der bestehenden
Compileumgebung
Erstellt auch Code 100
Kann NM**Cobol**, COBOL85, SCOBOL, TAL ...

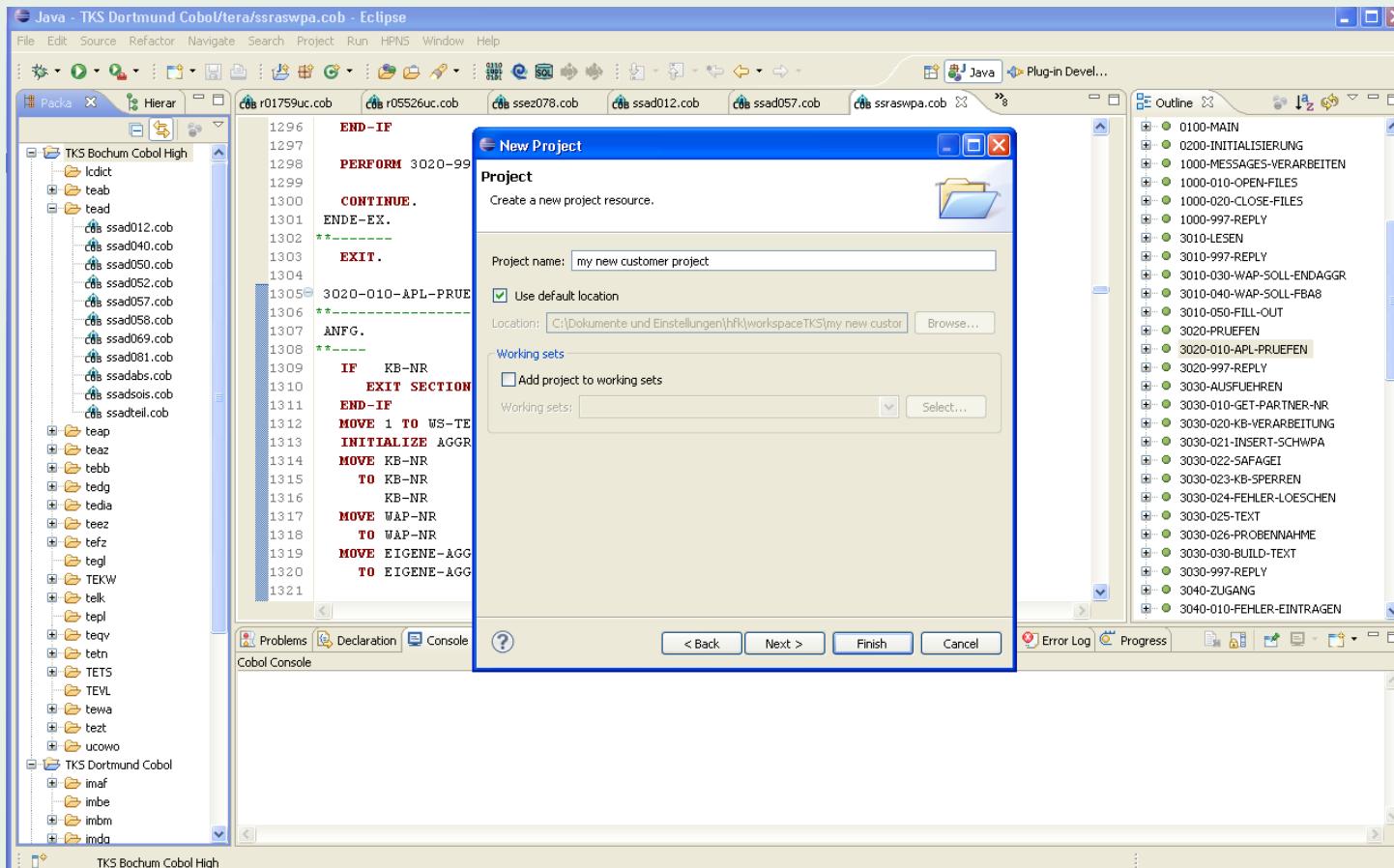
Aufbau CommitWork Developer Suite



Building a new project

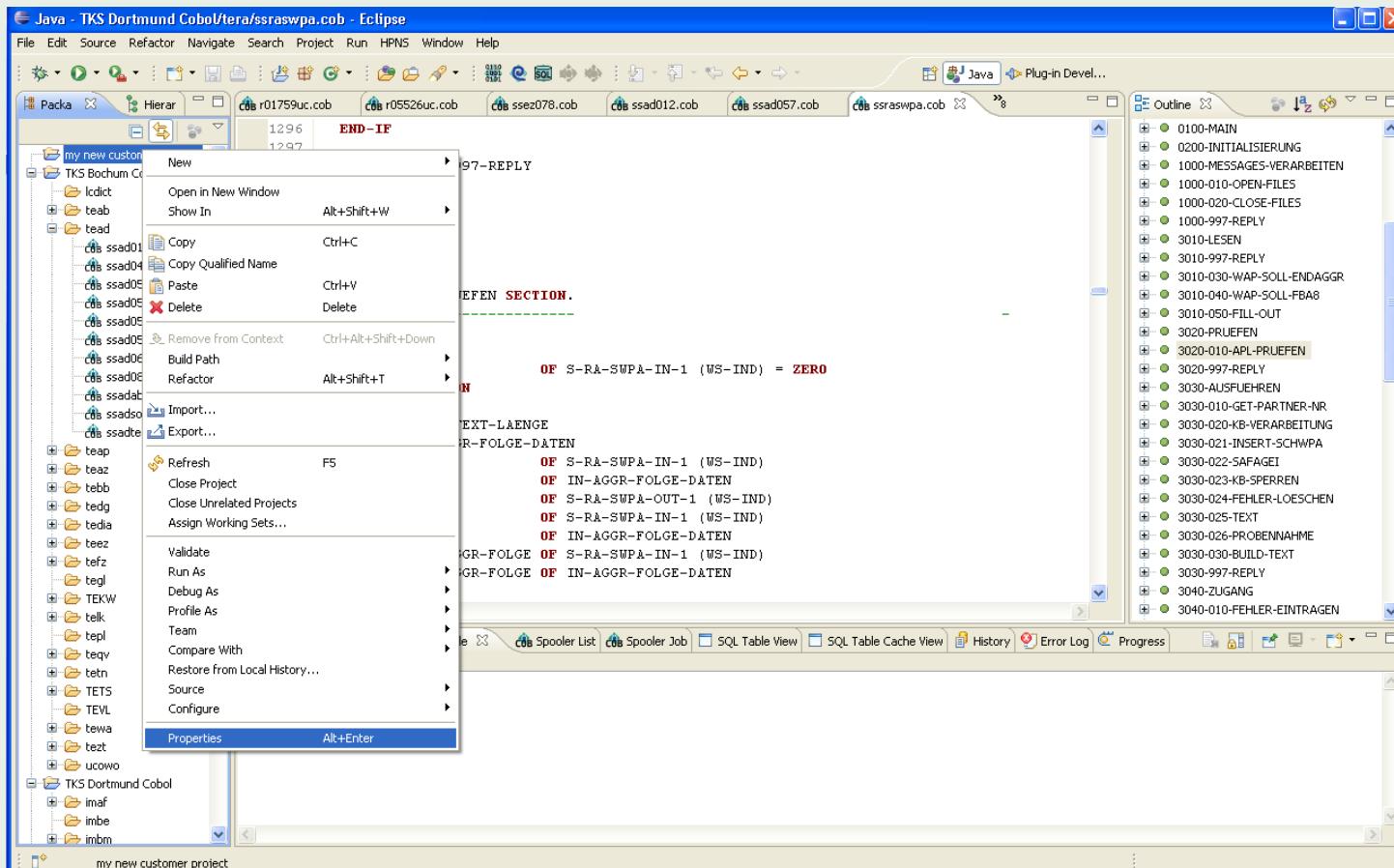


Building a new project



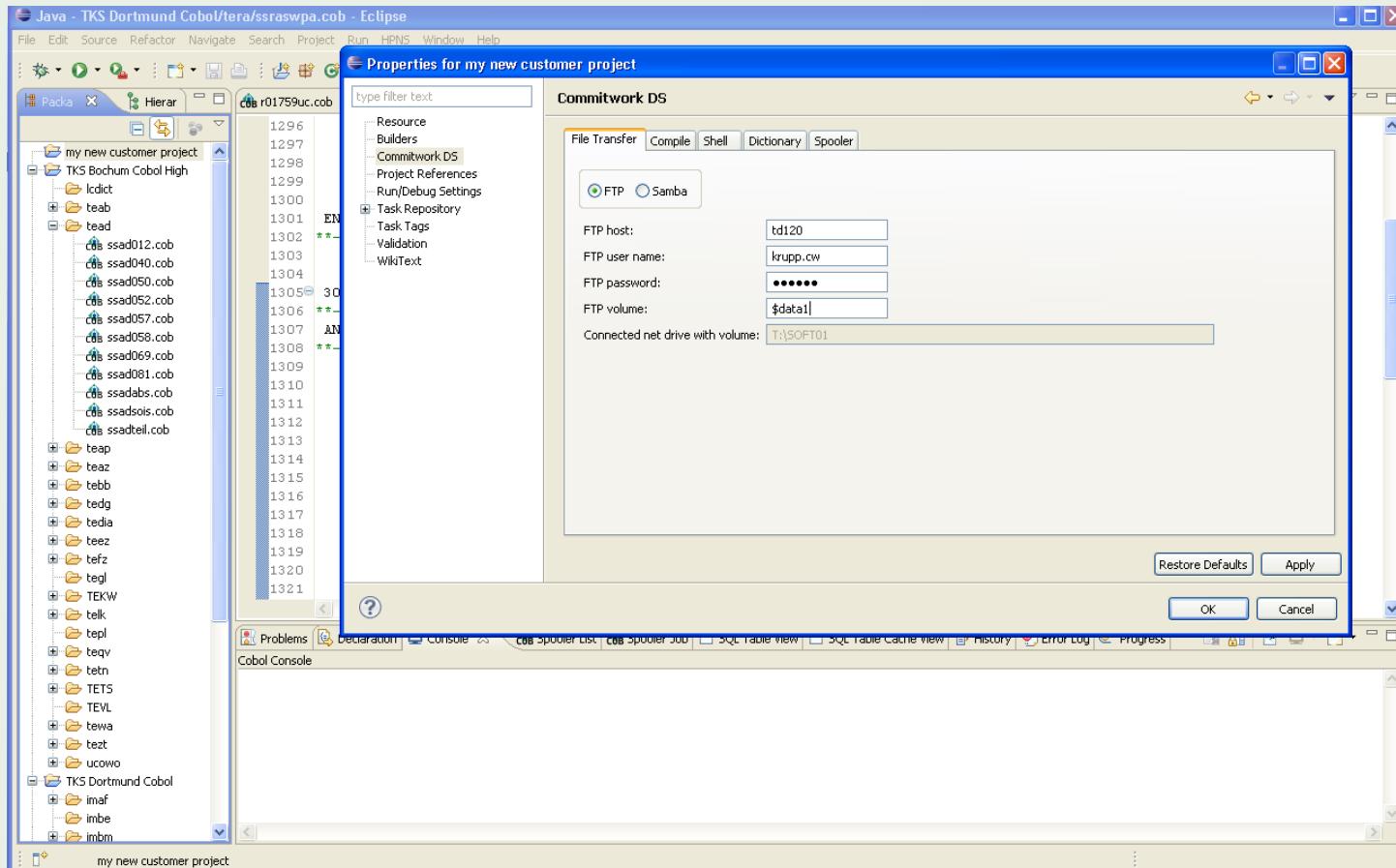
Give your project a name

Building a new project



Configuration of the properties

Building a new project

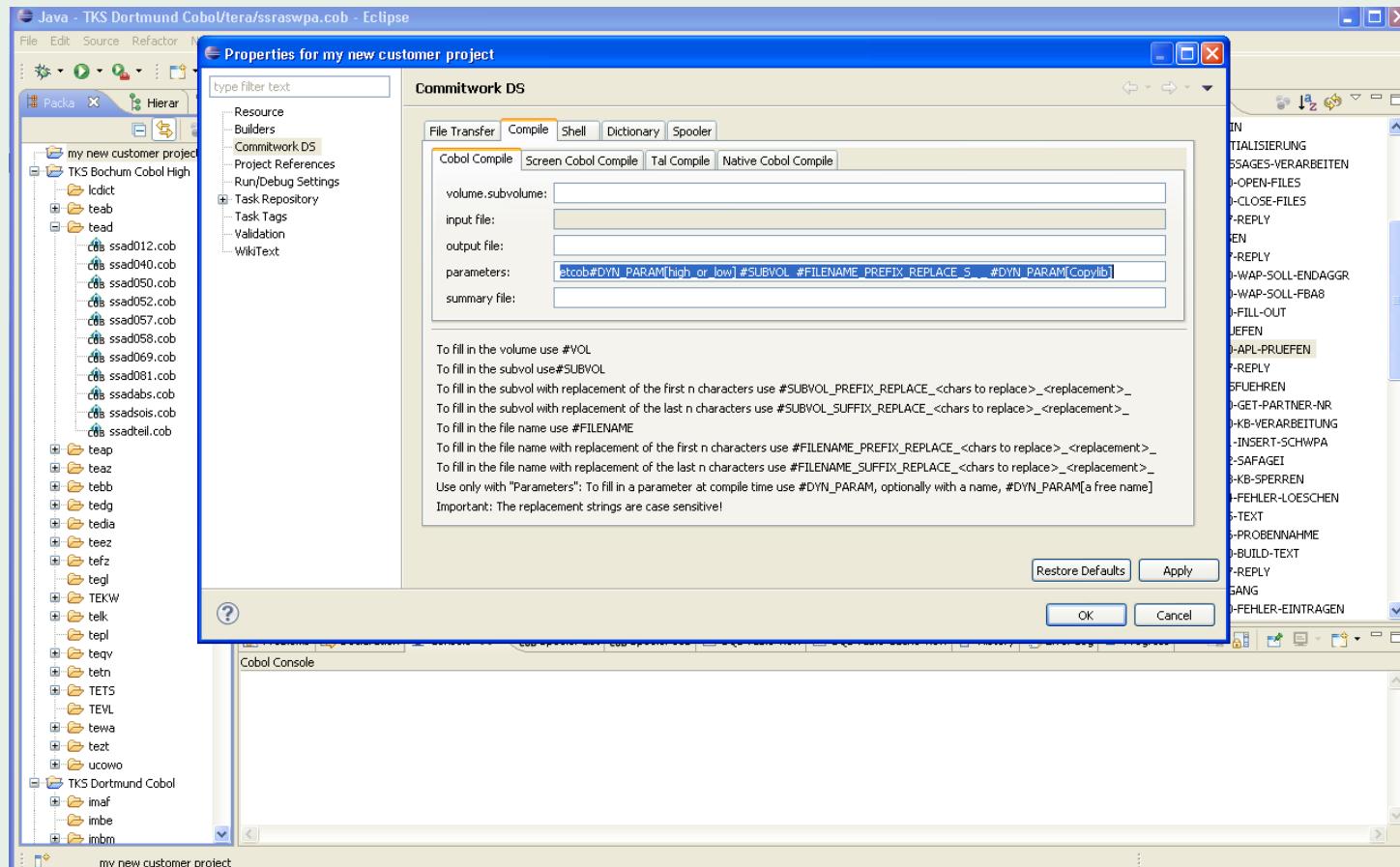


Configuration of the properties:

IP or DNS-Name, Guardian-User and Password

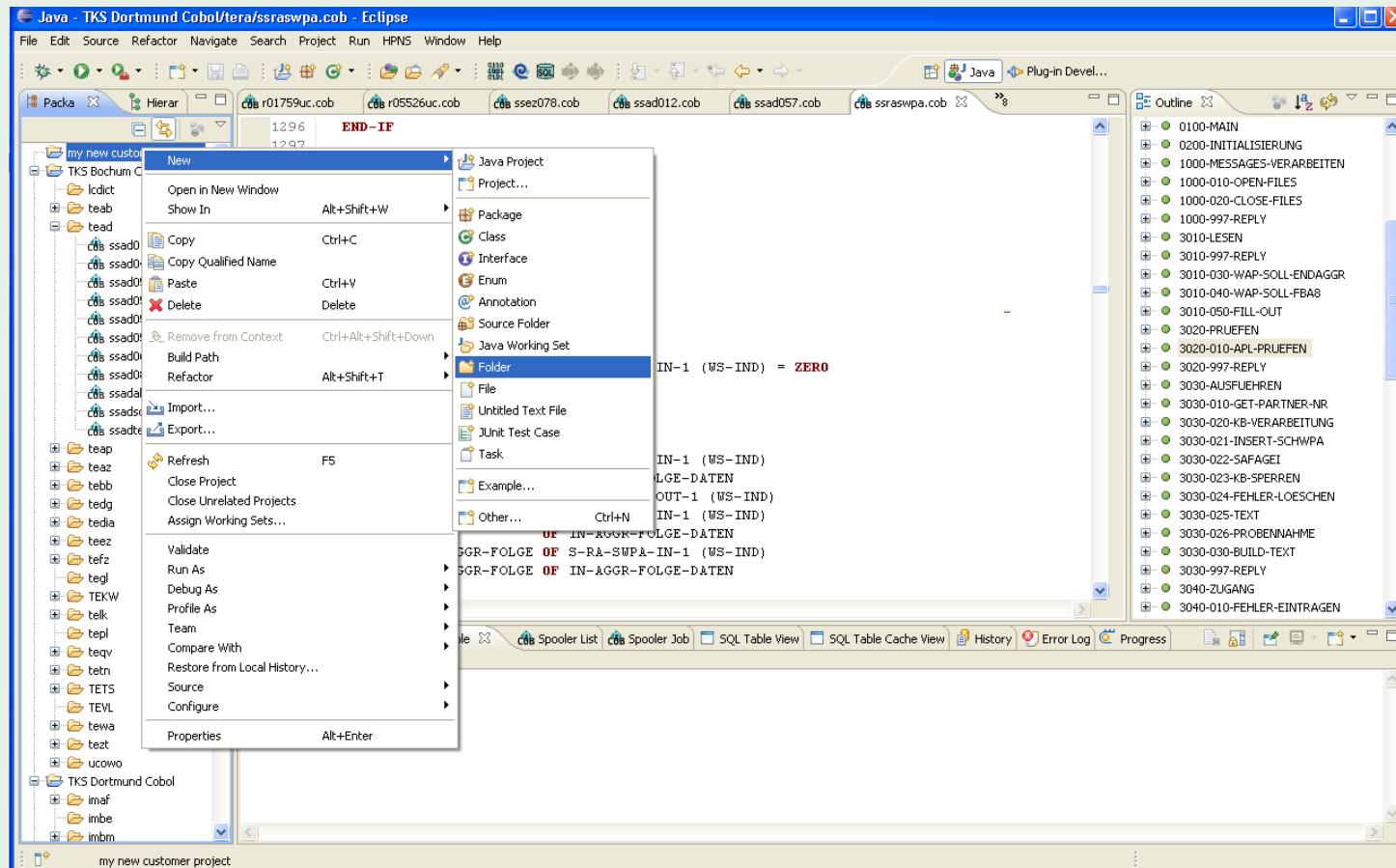
Guardian Default volume

Building a new project



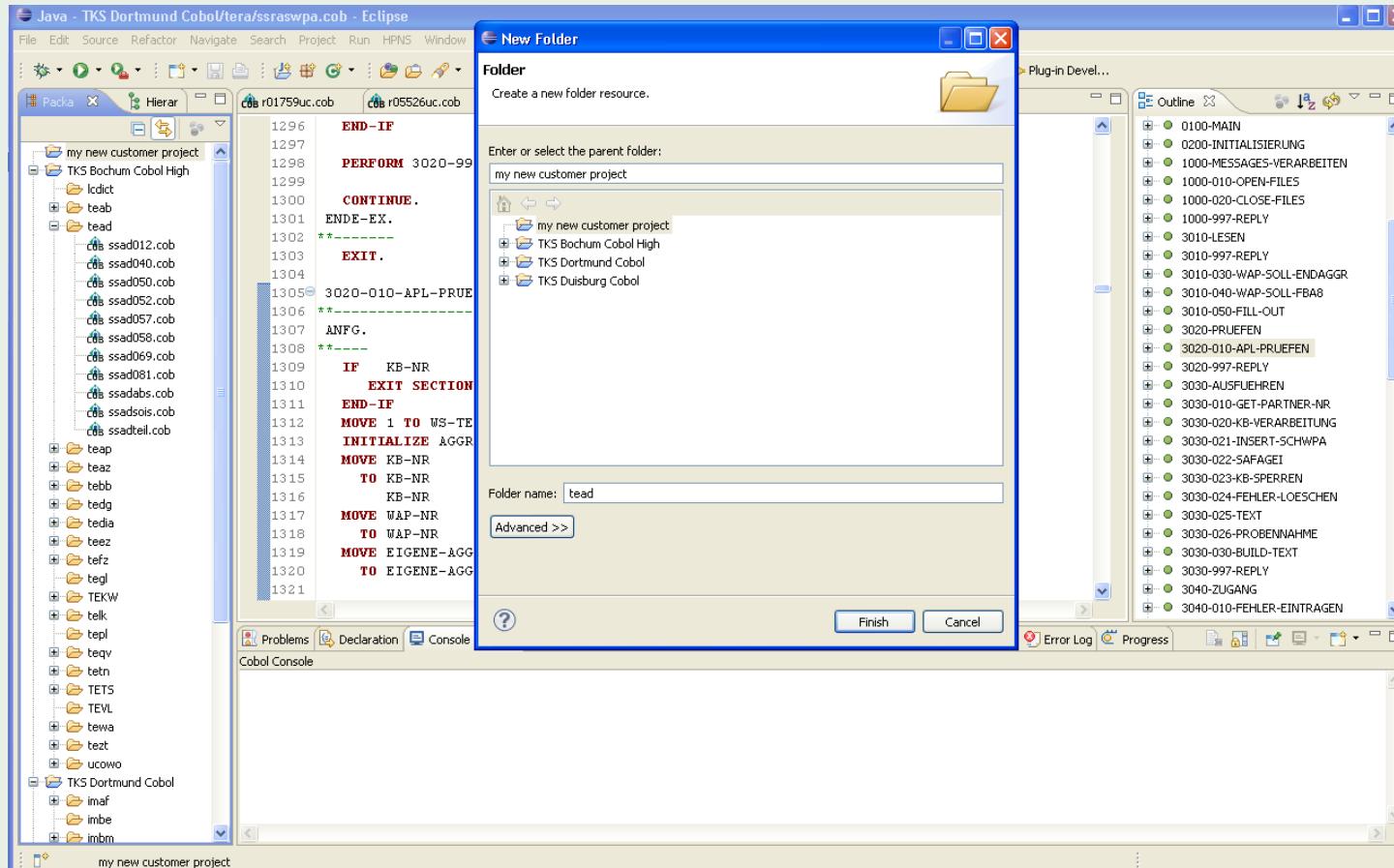
Configuration of the properties:
Compile command or macro

Building a new project



Adding a folder

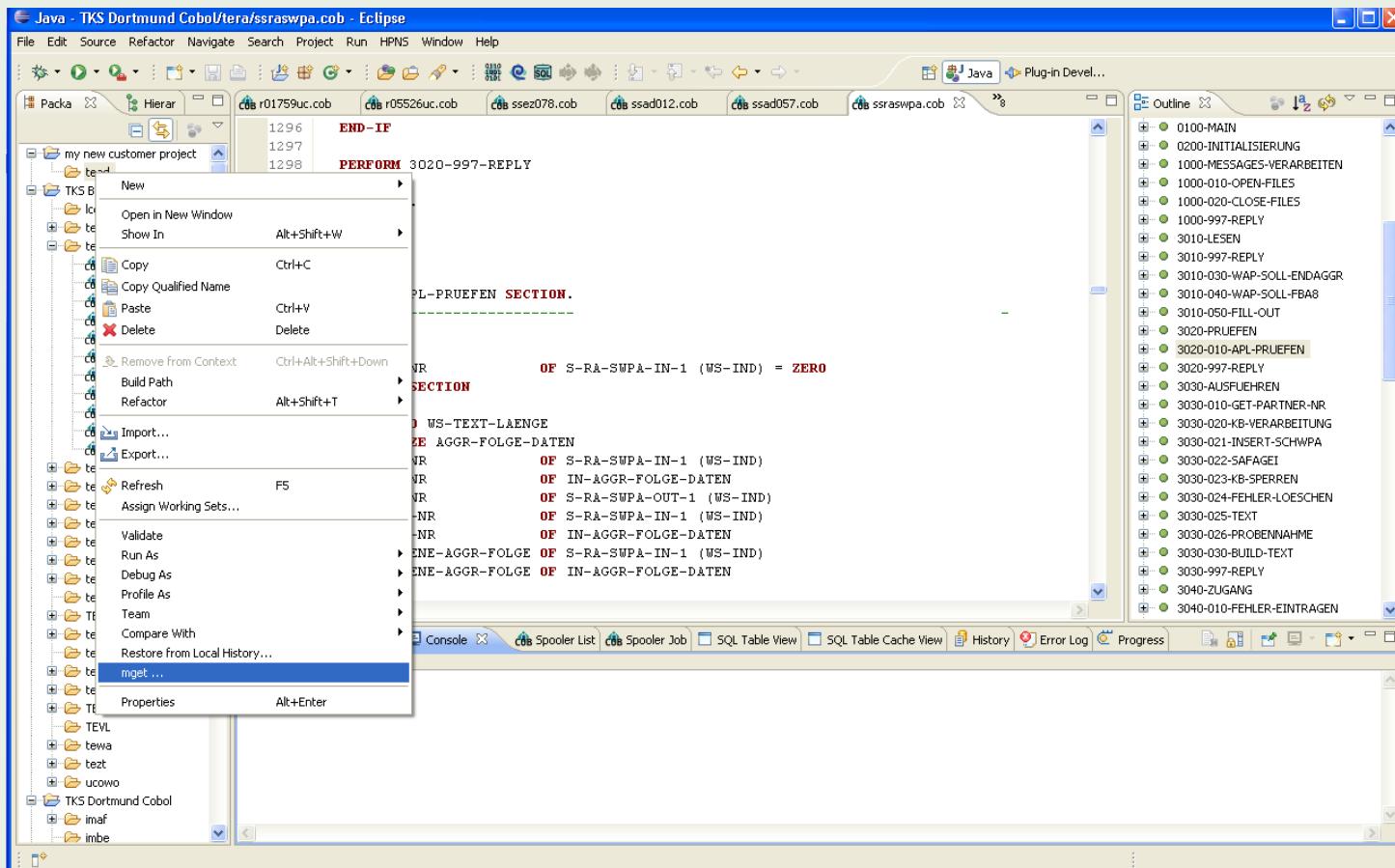
Building a new project



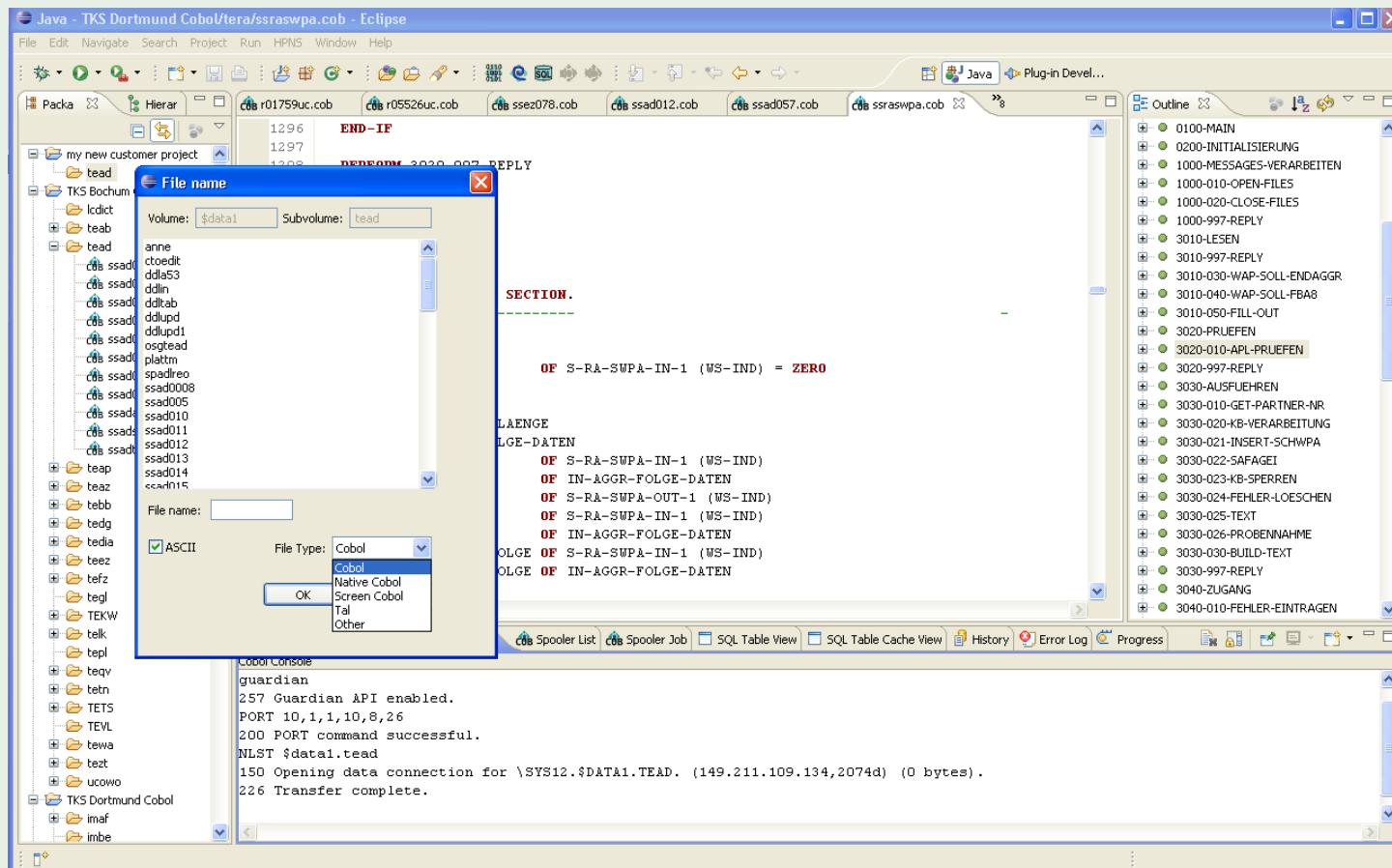
Building a new project

The Folder must have the same name as the subvolume on the
HP NSs

Getting program sources from the HP NS server

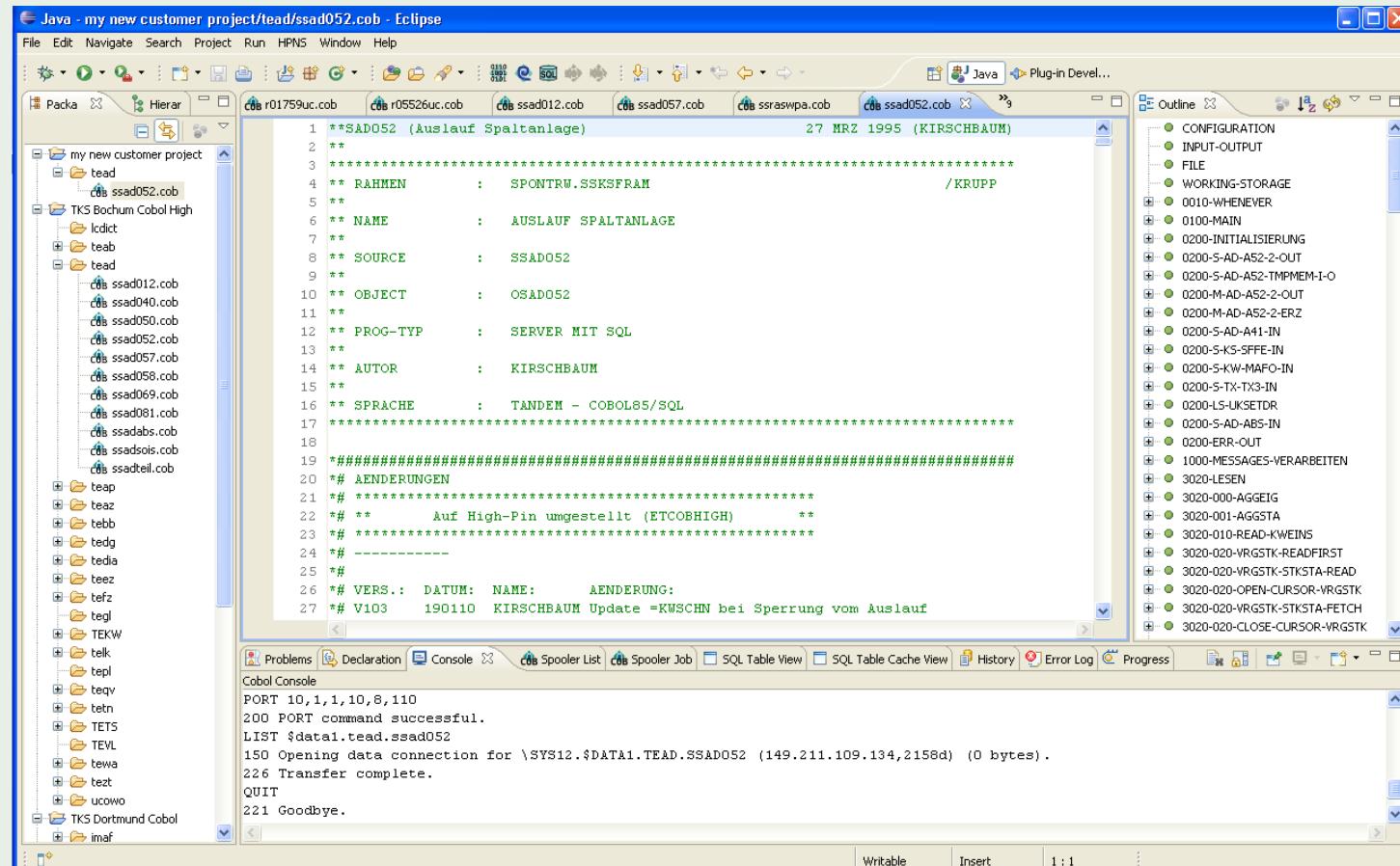


Getting program sources from the HP NS server



Evaluate the type,
Enter a wildcard or a filename

The Editor



The Editor

A screenshot of the Eclipse IDE interface, specifically the Cobol editor. The title bar reads "Java - my new customer project/tead/ssad052.cob - Eclipse". The left sidebar shows a project tree with several Cobol source files under "my new customer project/tead". The main editor area displays Cobol code, starting with section 3020-030-VRGSEQ-READ SECTION. The code includes various SQL statements and data definitions. A red circle highlights the right-hand margin of the editor, which contains a vertical list of code snippets or completion suggestions. The bottom of the screen shows the Eclipse status bar with "Writable", "Insert", and "1803 : 1". The bottom toolbar includes tabs for Problems, Declaration, Console, Spooler List, Spooler Job, SQL Table View, SQL Table Cache View, History, Error Log, and Process. The Cobol Console tab is active, showing a session log with commands like PORT, LIST, and QUIT.

Outline view

The Editor

The screenshot shows the Eclipse IDE interface with the Cobol Editor open. The editor displays Cobol source code for a program named ssad052.cob. A black arrow points from the Outline view on the right towards the editor window, indicating a shortcut or association between the two views.

```
5720 * Erzeugung bearbeiten
5721 * Mafo aufrufen
  PERFORM 3160-100-MAFO
  IF NOT WS-SF-CONTINUE
    PERFORM 3160-997-REPLY
    GO TO ENDE
  END-IF
* Korrektur des Sequenzgewichtes
  PERFORM 3020-030-VRGSEQ-READ
  IF NOT WS-SF-CONTINUE
    PERFORM 3160-997-REPLY
    GO TO ENDE
  END-IF
  PERFORM 3160-540-UPD-VRGSEQ
  IF NOT WS-SF-CONTINUE
    PERFORM 3160-997-REPLY
    GO TO ENDE
  END-IF
  IF KZ-IN-FERT-EINS OF M-AD-A52-2-IN (1) = "V"
    PERFORM 3160-152-PROTOKOLL
  *V84
  IF KZ-NACHMELDUNG OF M-AD-A52-2-OUT < 2
    AND AGGREGAT-IST OF M-AD-A52-2-OUT = "AGG542"
      PERFORM 3160-170-SKWEQUA
    END-IF
    IF WS-SF-CONTINUE
      SET KEIN-FOLGEPROGRAMM OF M-AD-A52-2-OUT TO TRUE
  END-IF
  
```

The Outline view on the right lists various Cobol statements and their corresponding line numbers, such as 0200-M-AD-A52-2-OUT, 0200-5-AD-A41-IN, and 3020-030-VRGSEQ-READ.

The Cobol Console at the bottom shows the following output:

```
Cobol Console
PORT 10,1,10,8,110
200 PORT command successful.
LIST $data1.tead.ssad052
150 Opening data connection for \SYS12.$DATA1.TEAD.SSAD052 (149.211.109.134,2158d) (0 bytes).
226 Transfer complete.
QUIT
221 Goodbye.
```

Shortcut from Outline into the Editor

Walkaround: Edit and Compile

The screenshot shows the Eclipse IDE interface with a Cobol project open. The left sidebar displays a file tree for 'my new customer project' containing various Cobol source files. The central editor pane shows Cobol code with several annotations:

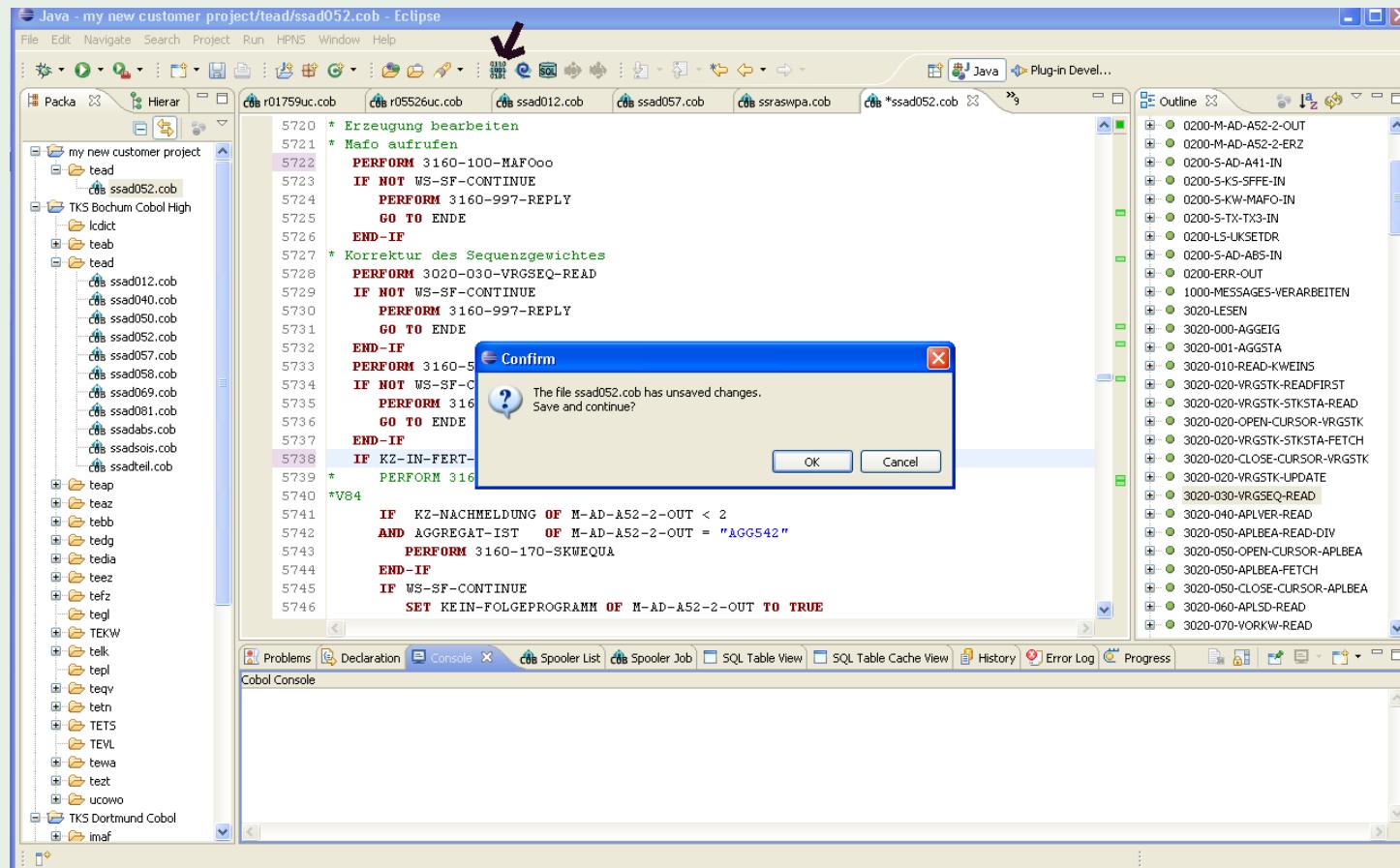
- A red arrow points to line 5722: `PERFORM 3160-100-MAFOoo`.
- A red arrow points to line 5738: `IF KZ-IN-FERT-EINS OF M-AD-A52-2-INoo (1) = "V"`.
- A green arrow points to line 5740: `*V84`.
- A red arrow points to line 5745: `SET KEIN-FOLGEPROGRAMM OF M-AD-A52-2-OUT TO TRUE`.

The bottom console window shows the output of a Cobol run:

```
Cobol Console
PORT 10,1,10,8,110
200 PORT command successful.
LIST $data1.tead.ssad052
150 Opening data connection for \SYS12.$DATA1.TEAD.SSAD052 (149.211.109.134,2158d) (0 bytes).
226 Transfer complete.
QUIT
221 Goodbye.
```

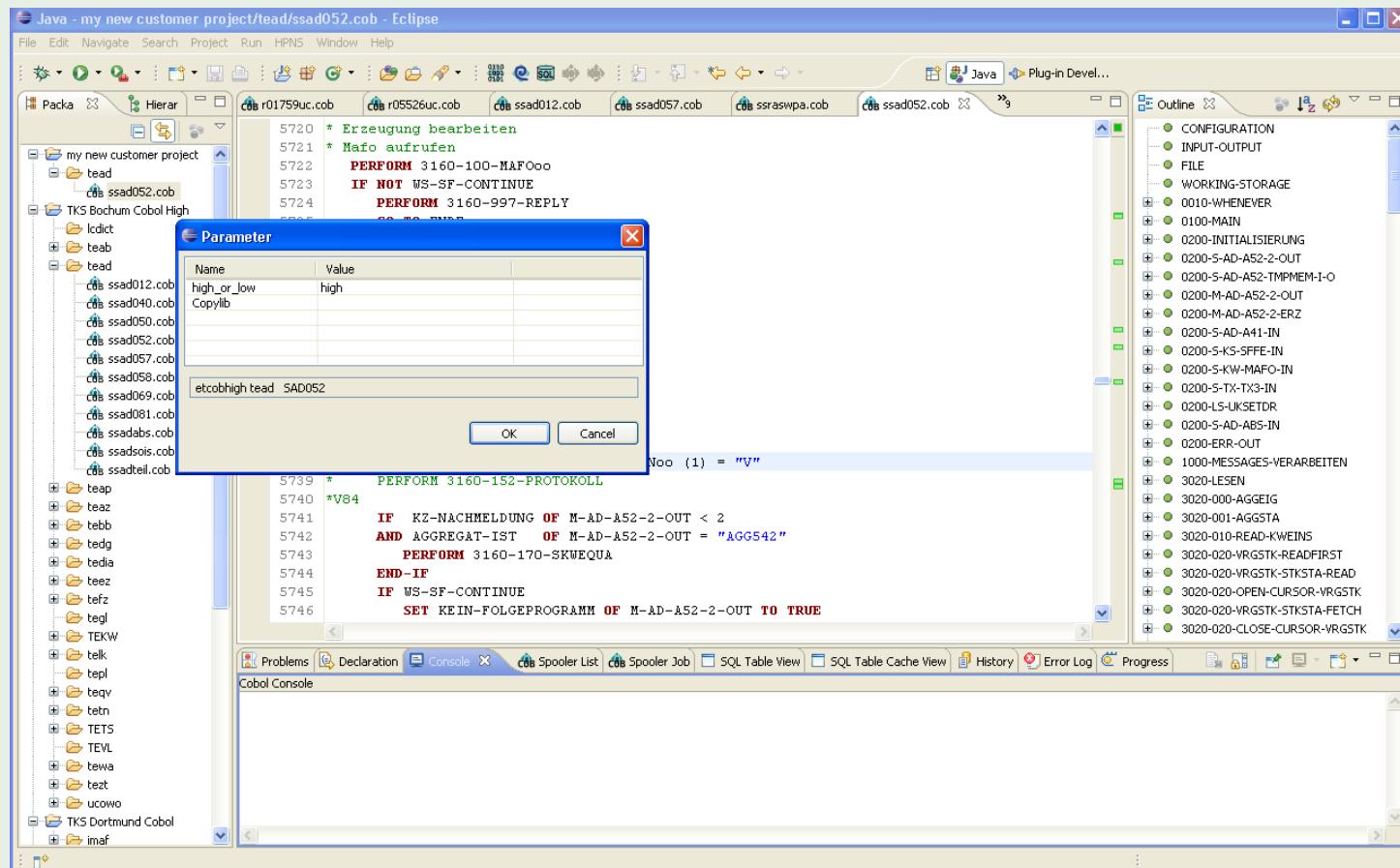
First make a mistake

Walkaround: Edit and Compile



Start the Compiler

Walkaround: Edit and Compile



When indicated, enter parameters

Walkaround: Edit and Compile

The screenshot shows the Eclipse IDE interface with the following details:

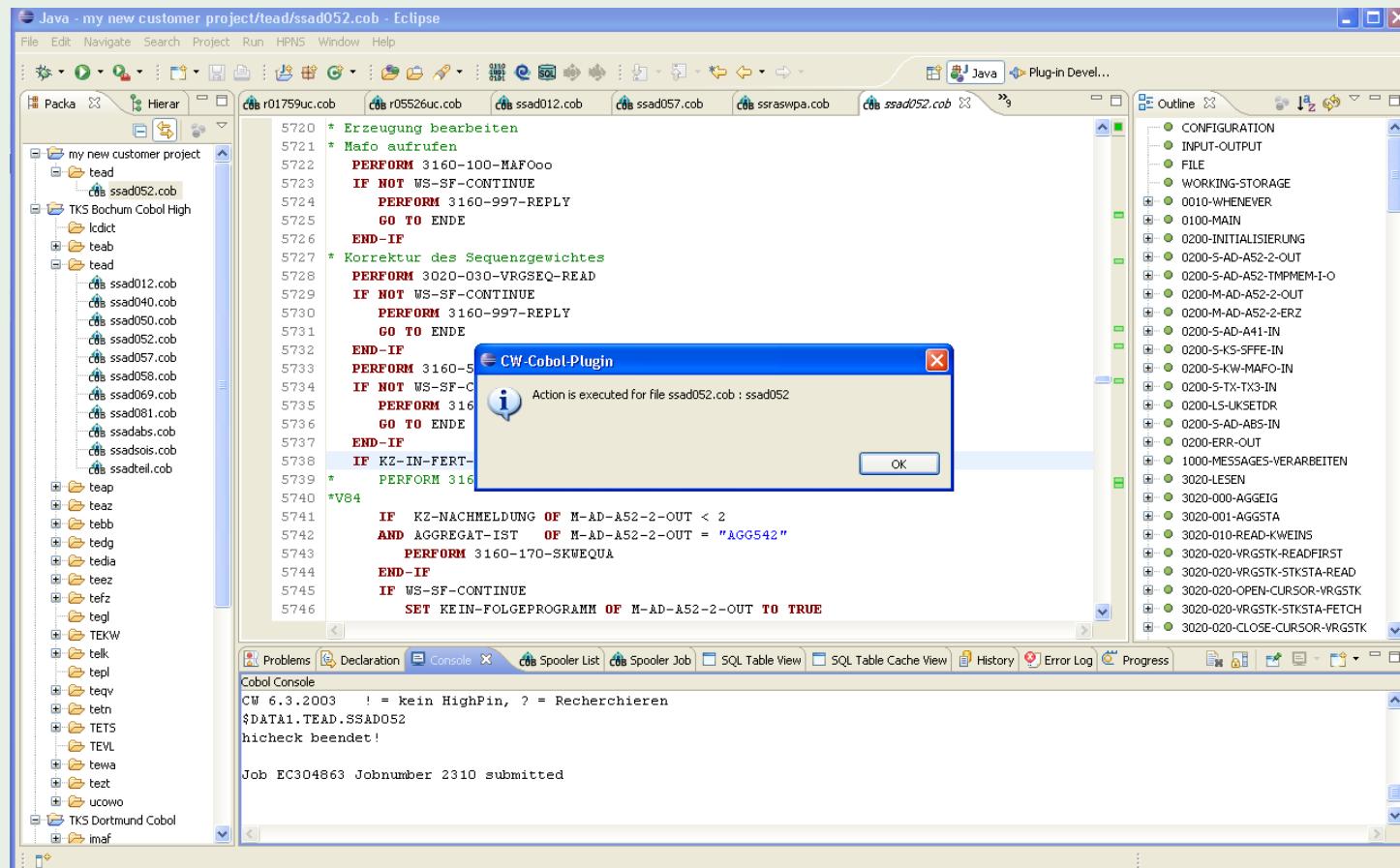
- Project Explorer (left):** Shows a project named "my new customer project" containing several COBOL source files: r01759uc.cob, r05526uc.cob, ssad012.cob, ssad057.cob, ssraswpa.cob, and ssad052.cob.
- Editor (center):** Displays COBOL code for "ssad052.cob". The code includes comments like "* Erzeugung bearbeiten" and "* Mafo aufrufen". It contains several PERFORM statements and IF NOT WS-SF-CONTINUE blocks. A specific section is highlighted with a blue background and labeled "*V84".
- Outline View (right):** Shows a tree view of the COBOL program structure, including sections like CONFIGURATION, INPUT-OUTPUT, FILE, WORKING-STORAGE, and various procedure division entries.
- Console (bottom):** Shows the output of a Cobol Console session:

```
226 Transfer complete.  
PORT 10,1,1,10,10,31  
200 PORT command successful.  
LIST $data1.tead.ssad052  
150 Opening data connection for \SYS12.$DATA1.TEAD.SSAD052 (149.211.109.134,2591d) (0 bytes).  
226 Transfer complete.
```

A red arrow points from the text "the compile request is running" to the progress bar at the bottom of the console window.
- Progress Bar (bottom right):** Shows "Compile Job: (0%)".

the compile request is running

Walkaround: Edit and Compile



The compile request was successfull

Walkaround: Edit and Compile

The screenshot shows the Eclipse IDE interface with several windows open:

- Java - my new customer project/tead/ssad052.cob - Eclipse**: The main editor window displaying Cobol code. A red arrow points to the line `SET KEIN-FOLGEPROGRAMM OF M-AD-A52-2-OUT TO TRUE`. The code is as follows:

```
5720 * Erzeugung bearbeiten
5721 * Mafo aufrufen
      PERFORM 3160-100-MAFOoo
      IF NOT WS-SF-CONTINUE
          PERFORM 3160-997-REPLY
          GO TO ENDE
      END-IF
      * Korrektur des Sequenzgewichtes
      PERFORM 3020-030-VRGSEQ-READ
      IF NOT WS-SF-CONTINUE
          PERFORM 3160-997-REPLY
          GO TO ENDE
      END-IF
      PERFORM 3160-540-UPD-VRGSEQ
      IF NOT WS-SF-CONTINUE
          PERFORM 3160-997-REPLY
          GO TO ENDE
      END-IF
      IF KZ-IN-FERT-EINS OF M-AD-A52-2-INoo (1) = "V"
          PERFORM 3160-152-PROTOKOLL
      *V84
          IF KZ-NACHMELDUNG OF M-AD-A52-2-OUT < 2
              AND AGGREGAT-IST OF M-AD-A52-2-OUT = "AGG542"
                  PERFORM 3160-170-SKWEQUA
              END-IF
          IF WS-SF-CONTINUE
              SET KEIN-FOLGEPROGRAMM OF M-AD-A52-2-OUT TO TRUE
      
```

- Hierarchy View**: Shows the project structure with files like r01759uc.cob, r05526uc.cob, ssad012.cob, ssad057.cob, ssraswpa.cob, and ssad052.cob.
- Outline View**: Displays a tree view of Cobol objects, including CONFIGURATION, INPUT-OUTPUT, FILE, WORKING-STORAGE, and various procedure division sections.
- Problems View**: Shows no errors or warnings.
- Console View**: Shows the output of the compilation process.
- Spooler List View**: A table showing spooler jobs. A red arrow points to the entry for job 2508. The table data is as follows:

| No | Report | Location | #Pages | Status | #Copies | Date |
|------|-----------------|-----------------|--------|--------|---------|---------------------|
| 2507 | KRUPP CW | #NBATCH | 2 | Ready | 1 | 15.02.2010 14:31:25 |
| 2506 | EC30463250 26 | #NBATCH | 298 | Ready | 1 | 15.02.2010 14:31:25 |
| 2508 | KRUPP CW | #\$AD052 COBOL | 295 | Ready | 1 | 15.02.2010 14:31:24 |
| 409 | KRUPP CW | #SZTFG02SQLCOMP | 15 | Ready | 1 | 12.02.2010 13:33:09 |
| 406 | KRUPP CW | #NBATCH | 2 | Ready | 1 | 12.02.2010 13:33:09 |
| 405 | EC323661250 26 | #NBATCH | 122 | Ready | 1 | 12.02.2010 13:33:09 |
| 408 | SZTFG0220100212 | #SZTFG02AXCEL | 1 | Ready | 1 | 12.02.2010 13:33:07 |

Calling the spooler

Interpretation of the compiler results

The screenshot shows the Eclipse IDE interface with several windows open:

- Project Explorer:** Shows the project structure with files like r01759uc.cob, r05526uc.cob, ssad012.cob, ssad052.cob, etc.
- Code Editor:** Displays Cobol source code. A specific section is highlighted in green:

```
5720 * Erzeugung bearbeiten
5721 * Mafo aufrufen
      PERFORM 3160-100-MAFOoo
      IF NOT WS-SF-CONTINUE
        PERFORM 3160-997-REPLY
        GO TO ENDE
      END-IF
      * Korrektur des Sequenzgewichtes
      PERFORM 3020-030-VRGSEQ-READ
      IF NOT WS-SF-CONTINUE
        PERFORM 3160-997-REPLY
        GO TO ENDE
      END-IF
      PERFORM 3160-540-UPD-VRGSEQ
      IF NOT WS-SF-CONTINUE
        PERFORM 3160-997-REPLY
        GO TO ENDE
      END-IF
      IF K2-IN-FERT-EINS OF M-AD-A52-2-INoo (1) = "V"
        PERFORM 3160-152-PROTOKOLL
      *VS4
      IF K2-NACHMELDUNG OF M-AD-A52-2-OUT < 2
        AND AGGREGAT-IST OF M-AD-A52-2-OUT = "AGG542"
        PERFORM 3160-170-SKWEQUA
      END-IF
      IF WS-SF-CONTINUE
        SET KEIN-FOLGEPROGRAMM OF M-AD-A52-2-OUT TO TRUE
```
- Outline View:** Shows the class hierarchy and various methods.
- Console:** Displays the compiler output. Two arrows point to the last page of the output:

```
COBOL85 - T9257H01 - (15 MAR 09)
No object file produced
Number of compiler errors = 2
Number of compiler warnings = 0
Maximum symbol table size = 418516 bytes
Last message on page 293
Elapsed time - 00:00:08
```

Shortcut to the last page of the compiler output

Interpretation of the compiler results

The screenshot shows the Eclipse IDE interface with a Cobol project open. The left sidebar displays a file tree with various Cobol source files and their dependencies. The main editor window shows Cobol code with several error markers (red X icons) indicating compilation issues. One specific error is highlighted: "Error 271 ** Undefined object reference D-A52-2-INoo (1) = "V"" at line 5738. The code snippet involves a PERFORM loop and an IF-THEN-ELSE structure. The Problems view at the bottom shows two errors listed.

| Description | Resource | Path | Locat... | Type |
|--|--------------------|------------|----------|---------|
| Errors (2 items) | | | | |
| ** Error 271 ** Undefined object refer ssad052.cob | /my new custome... | line 57... | | Problem |
| ** Error 382 ** Undefined procedure r ssad052.cob | /my new custome... | line 5722 | | Problem |

Shortcut to the compiler errors

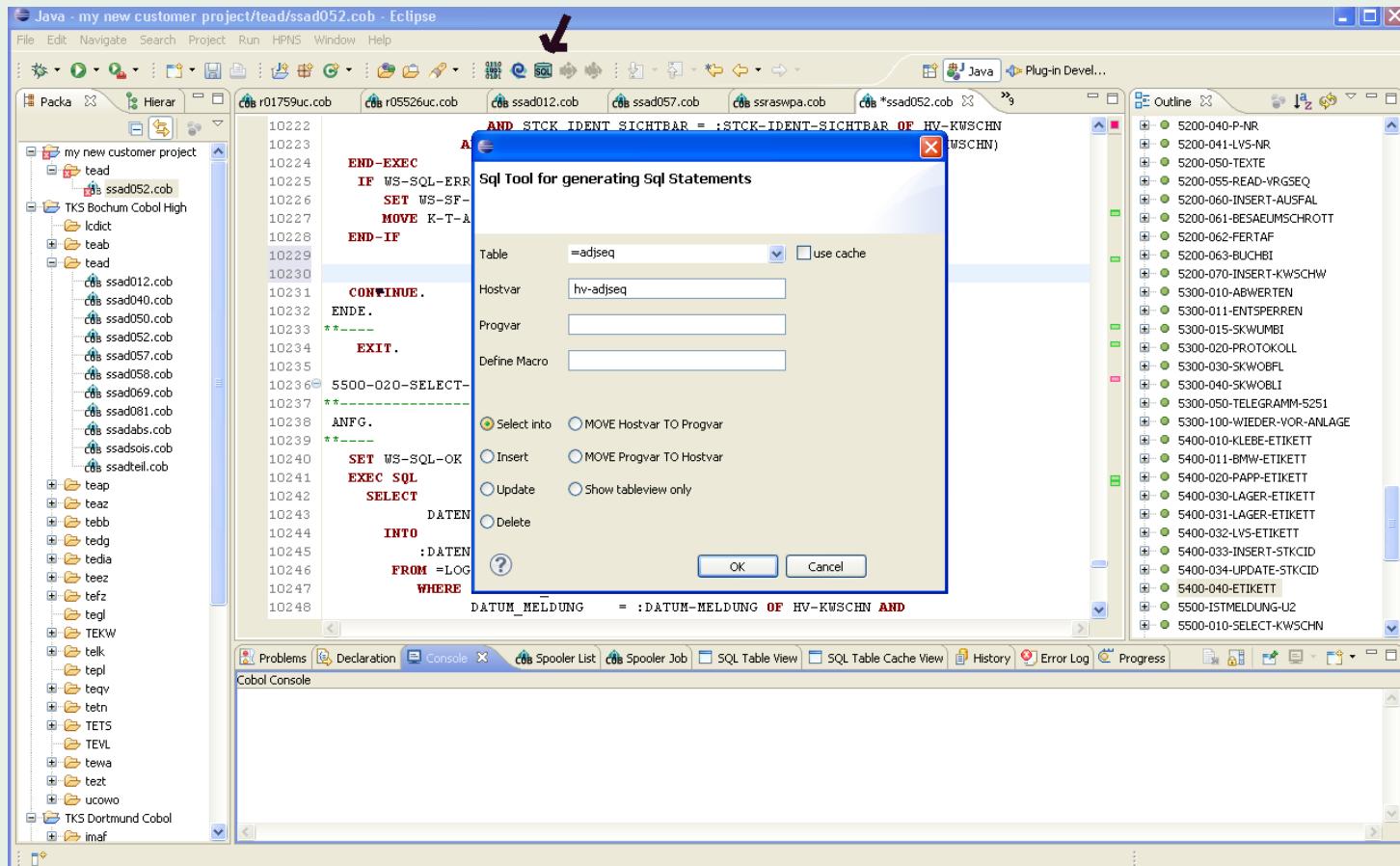
Interpretation of the compile results

The screenshot shows the Eclipse IDE interface with several windows open:

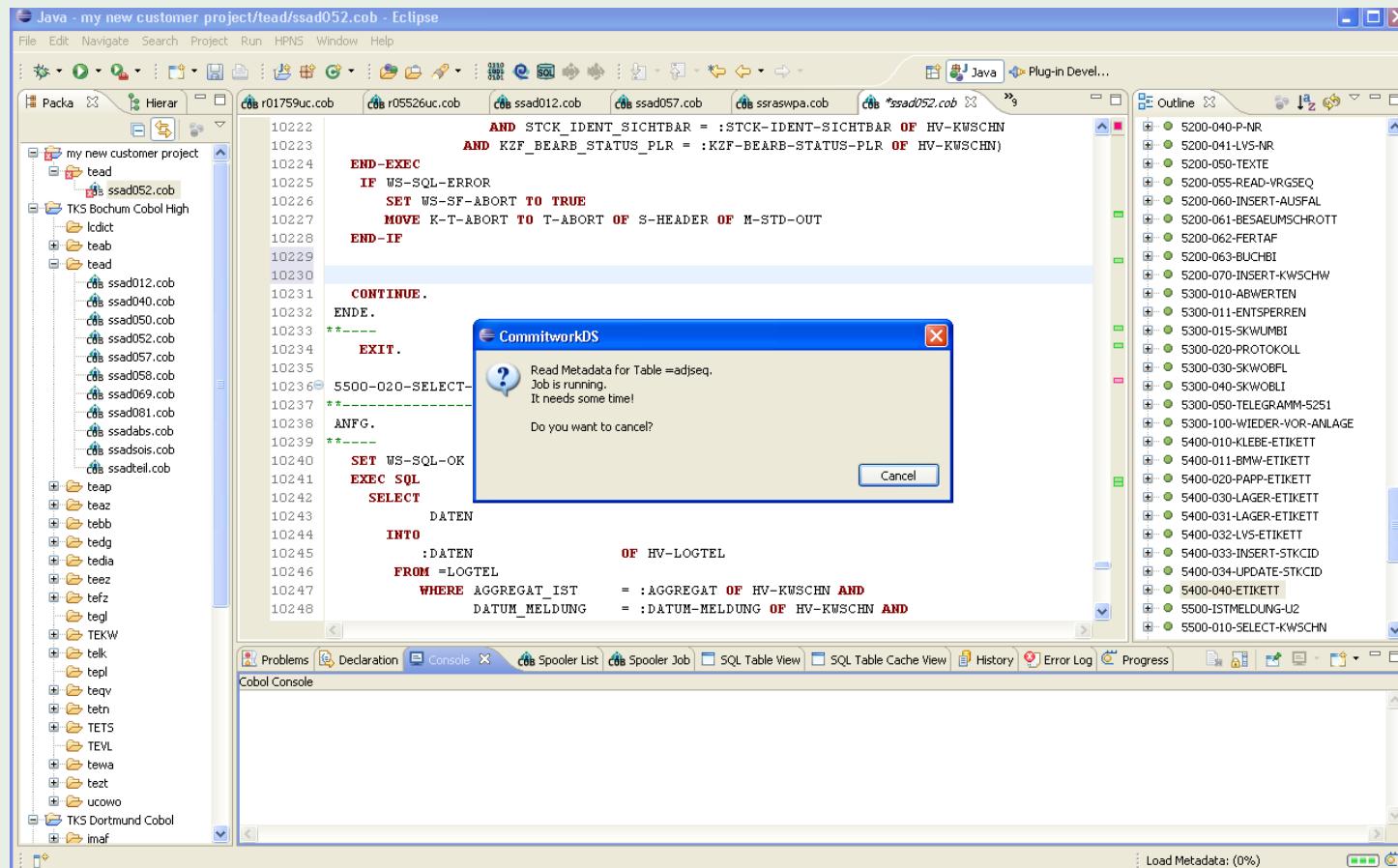
- Java - my new customer project/tead/ssad052.cob - Eclipse**: The main window displays Cobol source code for program `ssad052.cob`. The code includes various `PERFORM` statements, `IF` blocks, and comments. Some lines are marked with red error markers.
- Outline**: A tree view showing the structure of the Cobol program, including sections like `CONFIGURATION`, `INPUT-OUTPUT`, `FILE`, and `WORKING-STORAGE`.
- Problems**: Shows compilation errors and warnings.
- Console**: Displays the output of the COBOL compiler, including copyright information and default options.
- Spooler List**: Shows the spooler job details for the current session.
- Spooler Job**: Displays the full spooler job output, which includes the COBOL source code, compiler options, and the generated assembly-like code.

Alternatively browsing the complete spoolerjob

Generation of SQL-Statements



Generation of SQL-Statements



Accessing the SQL-catalog

Generation of SQL-Statements

The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer (left):** Shows the project structure under "my new customer project".
- Code Editor (center):** Displays Cobol code for generating SQL statements. The code includes:
 - Line 10225: IF WS-SQL-ERROR
 - Line 10226: SET WS-SF-ABORT TO TRUE
 - Line 10227: MOVE K-T-ABORT TO T-ABORT OF S-HEADER OF M-STD-OUT
 - Line 10228: END-IF
 - Line 10229: EXEC SQL
 - Line 10230: SELECT AGGREGAT_NAECHSTES, KZF_SEQ_ZUSTAND
 - Line 10231: ...
 - Line 10232: , ZEIT_DB_AEND, TIMESTAMP_DB_AEND
 - Line 10233: INTO AGGREGAT_NAECHSTES OF hv-adjseq
 - Line 10234: , KZF-SEQ-ZUSTAND OF hv-adjseq
 - Line 10235: ...
 - Line 10236: , ZEIT-DB-AEND OF hv-adjseq
 - Line 10237: , TIMESTAMP-DB-AEND OF hv-adjseq
 - Line 10238: FROM =adjseq
 - Line 10239: BROWSE ACCESS
 - Line 10240: END-EXEC.
 - Line 10241: CONTINUE.
 - Line 10242: ENDE.
 - Line 10243: ***
 - Line 10244: EXIT.
- Outline View (right):** Shows a tree view of generated SQL statements, including:
 - 5200-040-P-NR
 - 5200-041-LV5-NR
 - 5200-050-TEXT
 - 5200-055-READ-VRGSEQ
 - 5200-060-INSERT-AUSFAL
 - 5200-061-BESAUEUMSCHROTT
 - 5200-062-FERTAF
 - 5200-063-BUCHBI
 - 5200-070-INSERT-KW5CHW
 - 5300-010-ABWERTEN
 - 5300-011-ENTSPERREN
 - 5300-015-SKWUMBI
 - 5300-020-PROTOKOLL
 - 5300-030-SKWOBFL
 - 5300-040-SKWOBLI
 - 5300-050-TELEGRAMM-5251
 - 5300-100-WIEDER-VOR-ANLAGE
 - 5400-010-KLEBE-ETIKETT
 - 5400-011-BMW-ETIKETT
 - 5400-020-PAPP-ETIKETT
 - 5400-030-LAGER-ETIKETT
 - 5400-031-LAGER-ETIKETT
 - 5400-032-LVS-ETIKETT
 - 5400-033-INSERT-STKCID
 - 5400-034-UPDATE-STKCID
 - 5400-040-ETIKETT
 - 5500-1STMELDUNG-U2
 - 5500-010-SELECT-KW5CHN
- Bottom Bar:** Shows tabs for Problems, Declaration, Console, Spooler List, Spooler Job, SQL Table View, SQL Table Cache View, History, Error Log, and Progress. The SQL Table View tab is selected.
- Bottom Status Bar:** Shows Writable, Insert, and line number 10232 : 10.

Insert the result into the programm source

Remark: normally the `whereclause` with the primary-key is generated, but in this example the table is relative.



und, zuviel versprochen