



SOFORTE
SOLUTIONS FOR TEAMS

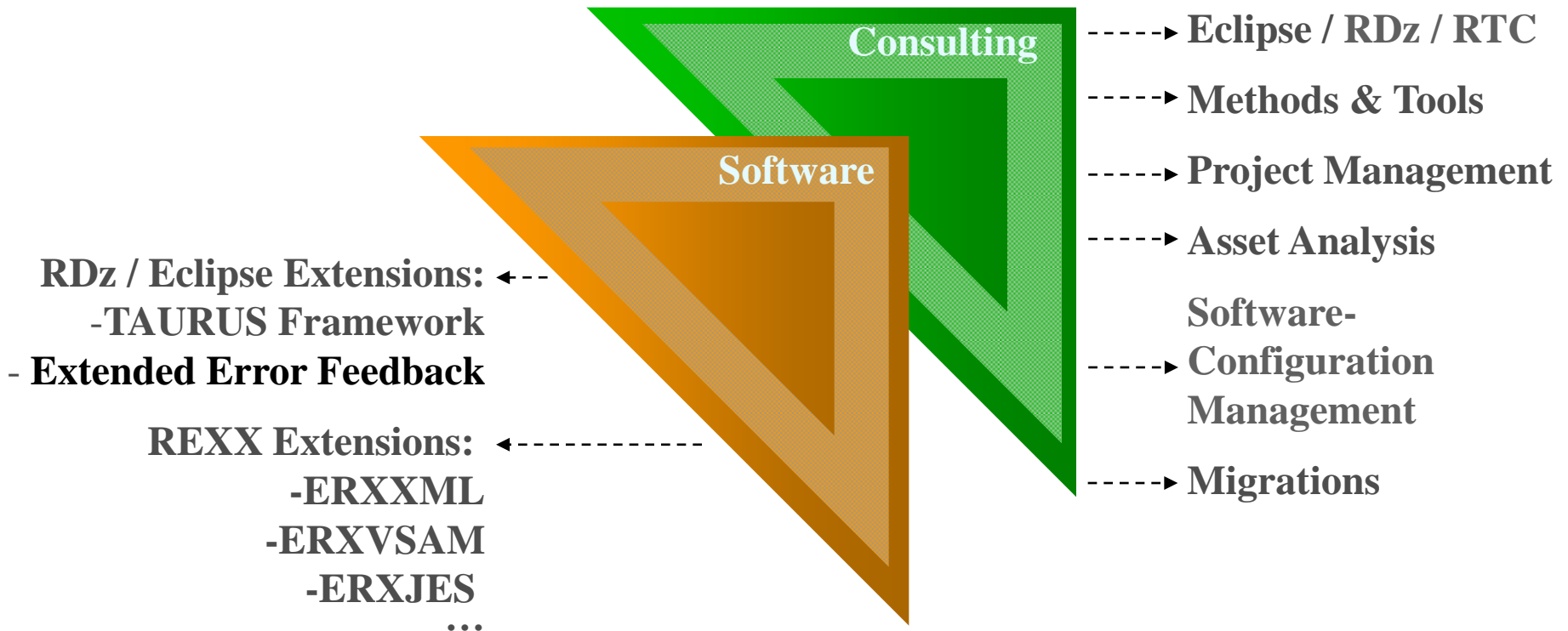
RDz – Extended Error Feedback Support

GSE Enterprise Modernization Conference
October 27, 2011

Anthony Rudd
a.rudd@soforte.de
<http://www.soforte.com>

SoforTe – Solutions for Teams

Your specialist for Enterprise Development Processes



Traditional ISPF-based development processing steps

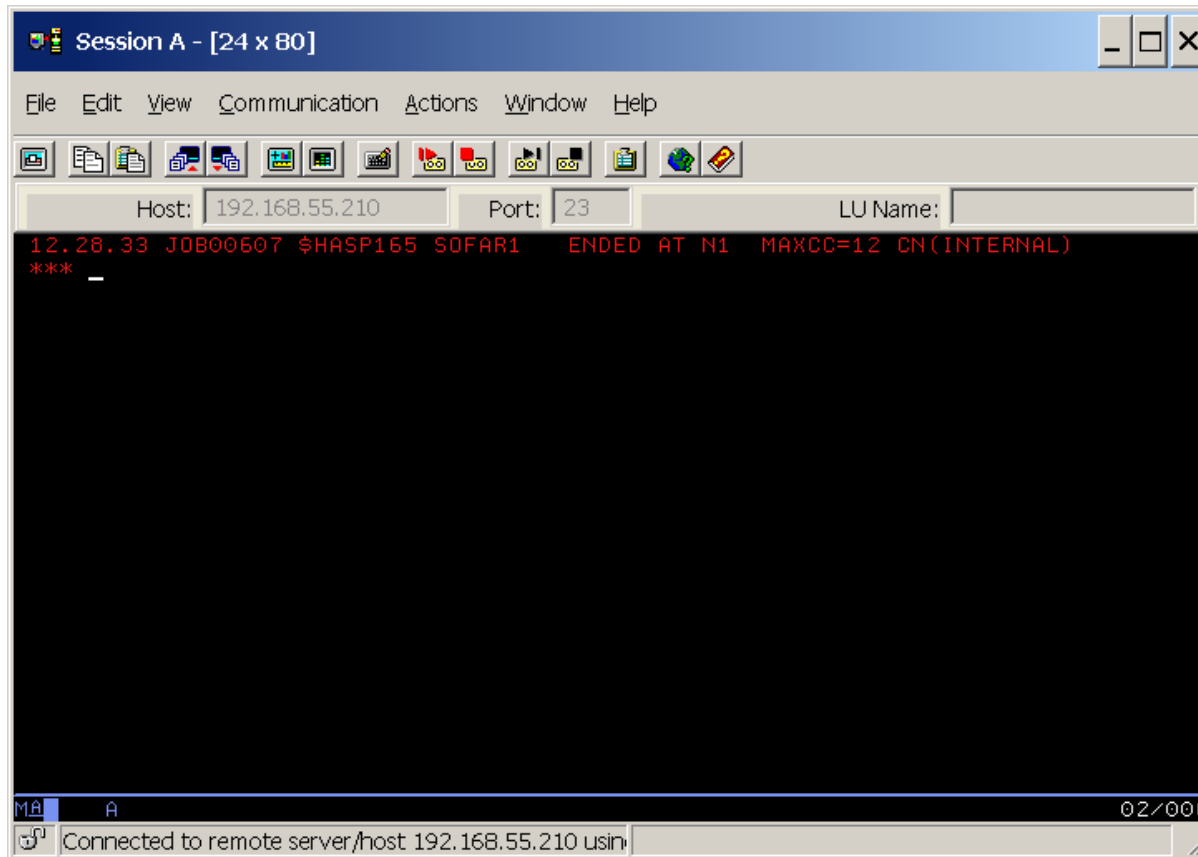


Traditional ISPF-based development



SOFORTE
SOLUTIONS FOR TEAMS

ISPF provides some information directly (via NOTIFY)
The shown maximum step completion code indicates whether a further analysis is necessary (MAXCC > 4).

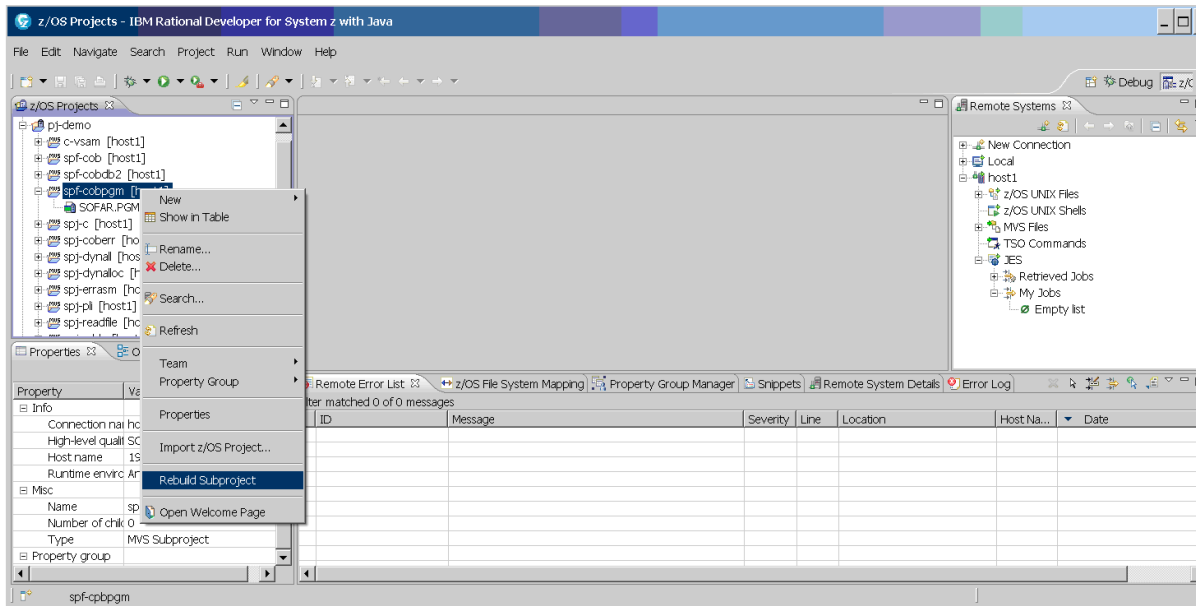


RDz-based development - introduction



SOFORTE
SOLUTIONS FOR TEAMS

RDz error feedback support is provided only for Rebuild (Sub)Project.



RDz-based development

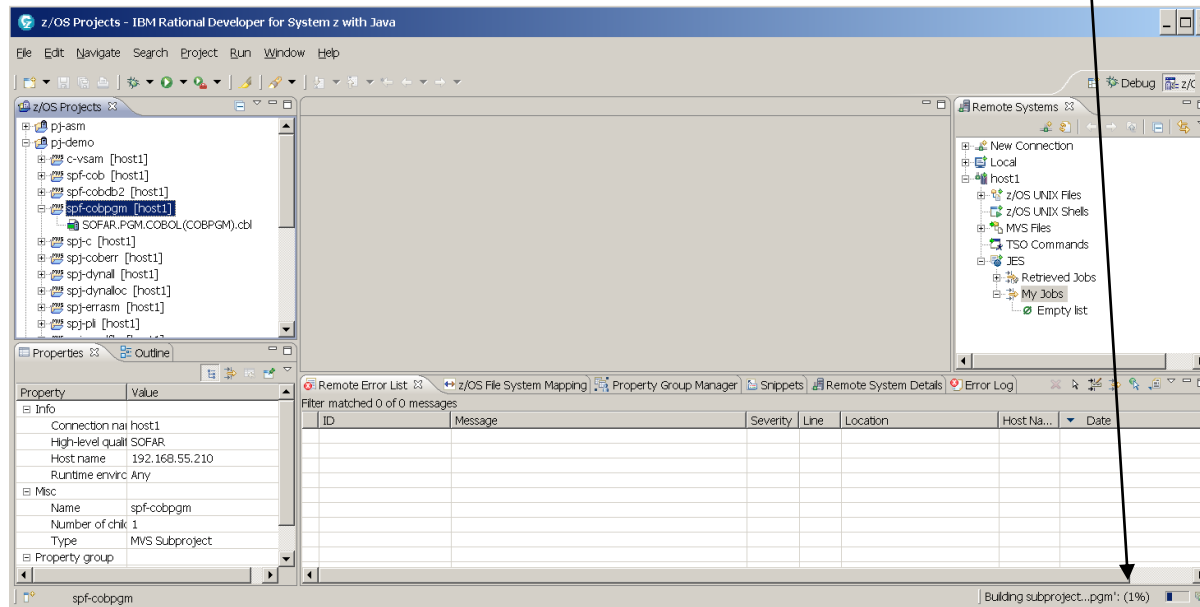


Progress bar indicates the status.

Optionally, the build job can be set to run in background, either by enabling the "Always run in background" checkbox

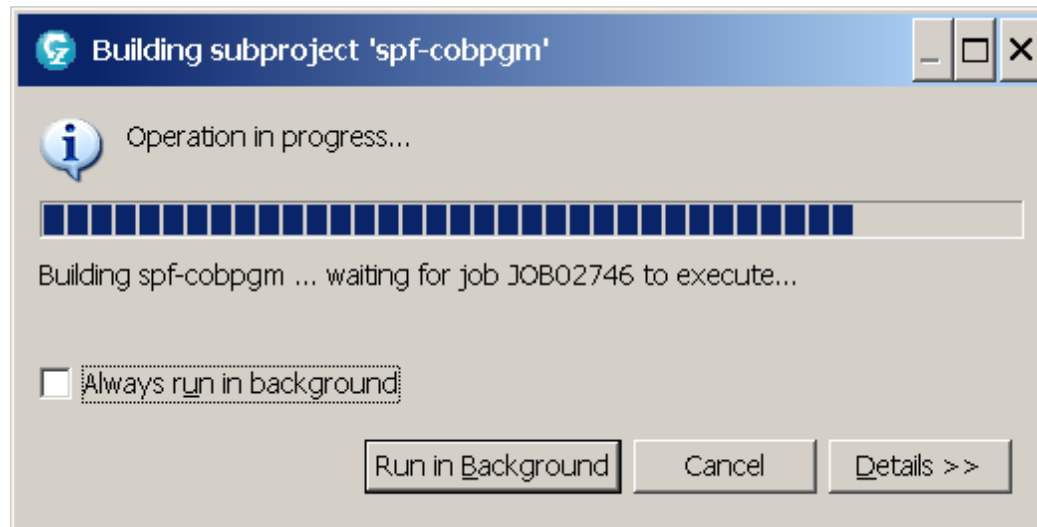
or by clicking the "Run in Background" key.

In this case, the job progress is shown as percentage in the lower right-hand frame.



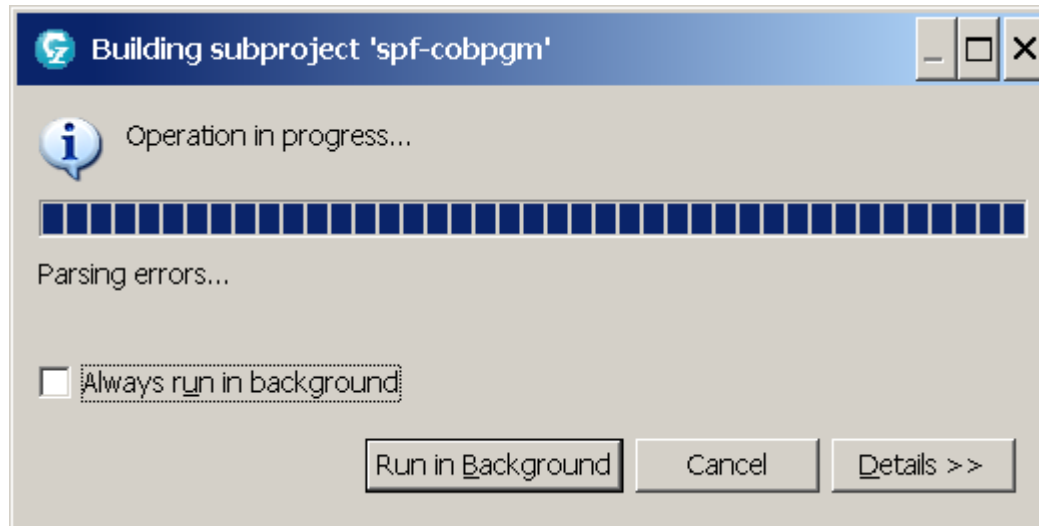
RDz-based development

When a build job is started, the associated job number is displayed in a pop-up window.



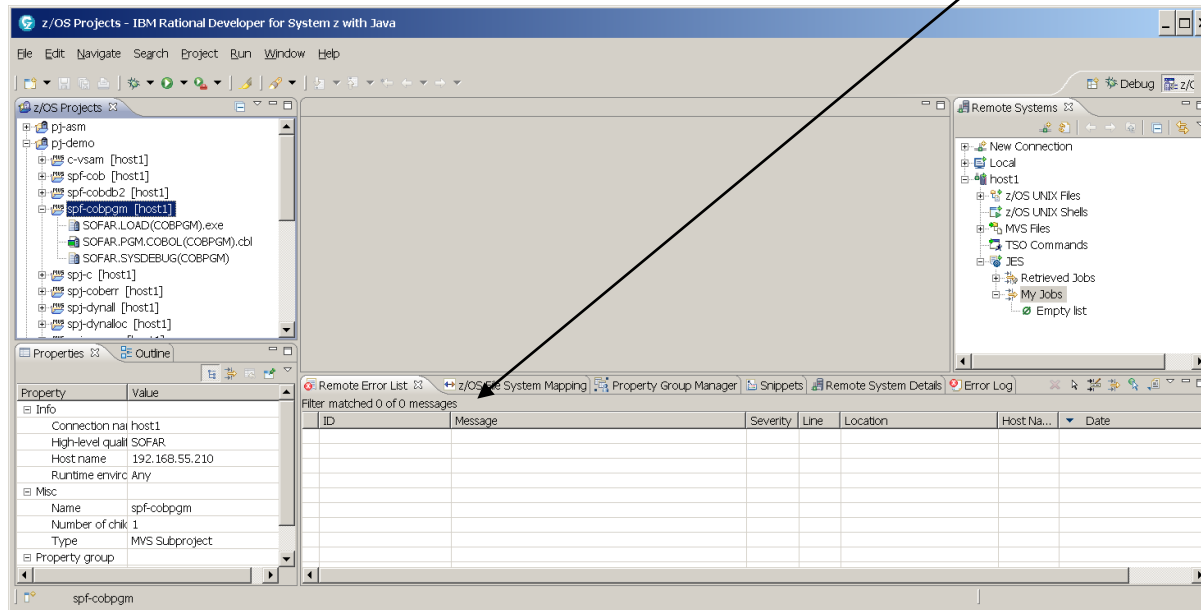
RDz-based development

On completion of the job, the compiler output is parsed for errors.



RDz-based development

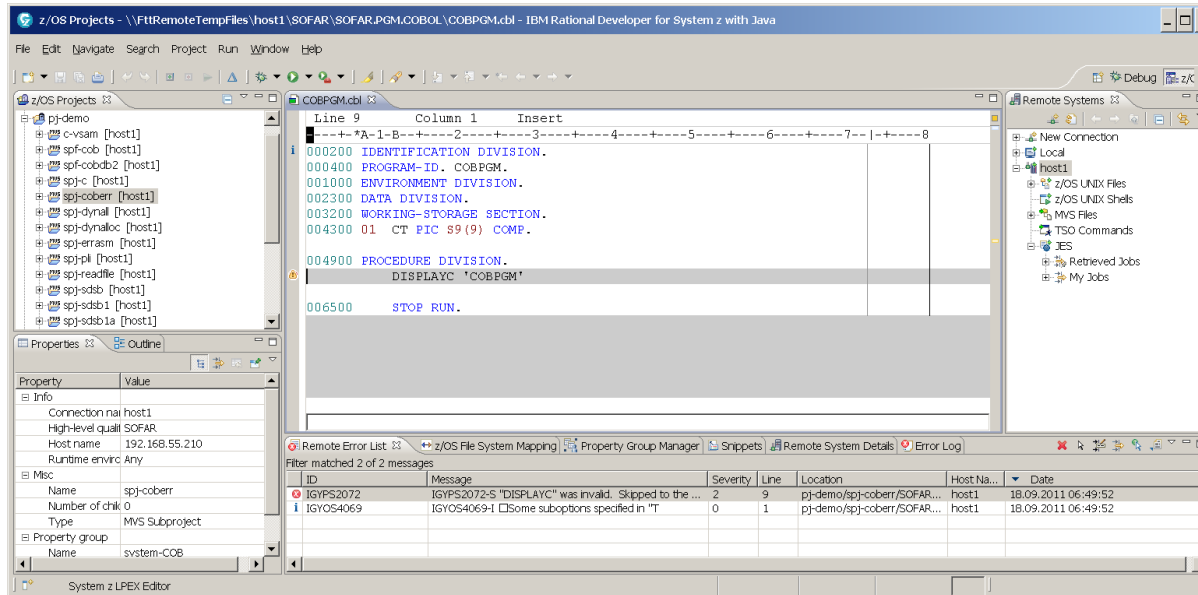
On completion of the parsing, any compilation errors will be displayed in the Remote Error List. Because there is no positive acknowledgment, the Remote Error List may be empty.



RDz-based development

Typical error feedback processing

Standard RDz error feedback for C/C++, COBOL, PLI compilers (optional, with CICS/DB2 coprocessor). Compiler errors are added to the Remote Error List; a double-click on an error message opens the editor with cursor positioned at the error line in the source program. The source program can be edited directly.



The screenshot displays the IBM Rational Developer for System z with Java interface. The main editor shows a COBOL program named COBPGM.cbl. The code is as follows:

```
Line 9      Column 1      Insert
-----*A-1-B-----2-----3-----4-----5-----6-----7-|-+-----8
000200  IDENTIFICATION DIVISION.
000400  PROGRAM-ID, COBPGM.
001000  ENVIRONMENT DIVISION.
002300  DATA DIVISION.
003200  WORKING-STORAGE SECTION.
004300  01  CT PIC S9(9) COMP.

004900  PROCEDURE DIVISION.
        DISPLAY 'COBPGM'

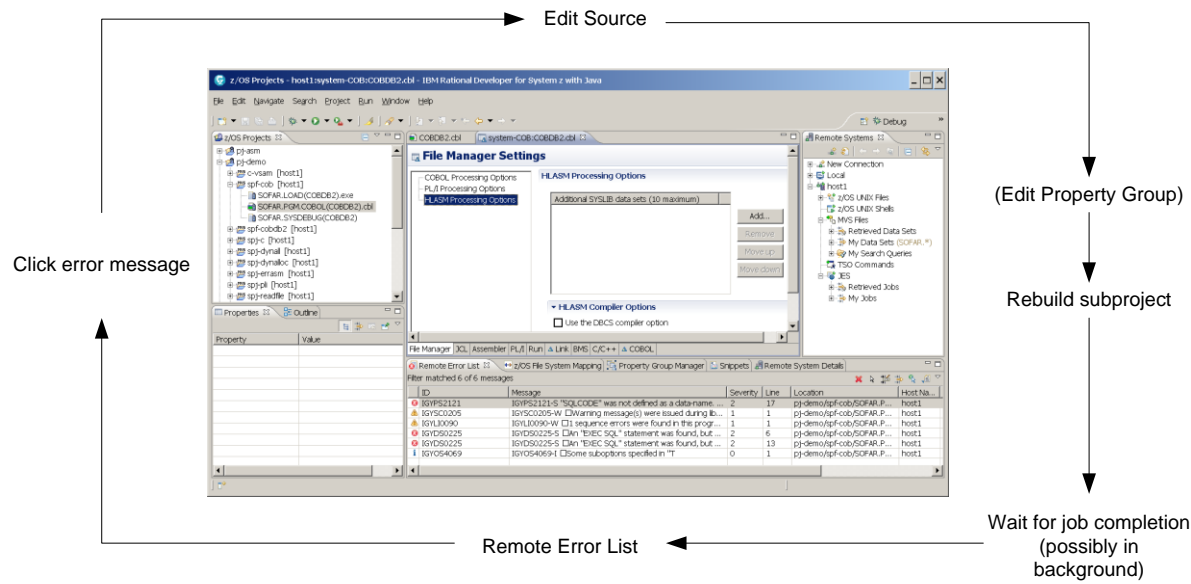
006500  STOP RUN.
```

The Remote Error List at the bottom shows two messages:

ID	Message	Severity	Line	Location	Host Na...	Date
IGYPS2072	IGYPS2072-S "DISPLAY" was invalid. Skipped to the ...	2	9	pj-demo/spj-coberr/SOFAR...	host1	18.09.2011 06:49:52
IGYOS4069	IGYOS4069-I □Some suboptions specified in "T	0	1	pj-demo/spj-coberr/SOFAR...	host1	18.09.2011 06:49:52

RDz-based development

Compile error processing steps

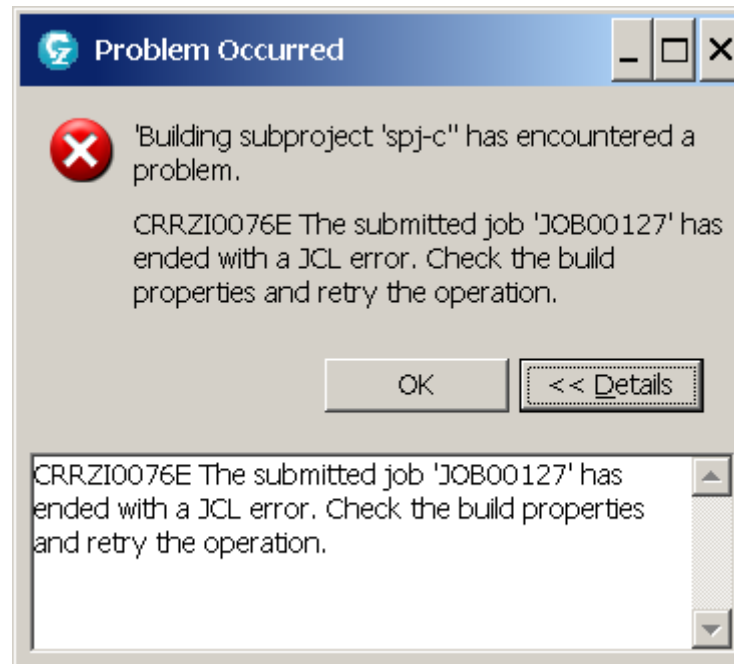


Problem: Errors in other steps remain undetected.

RDz-based development

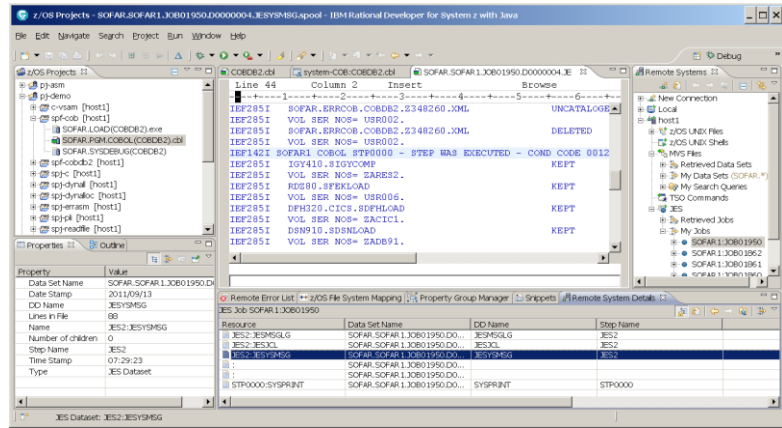
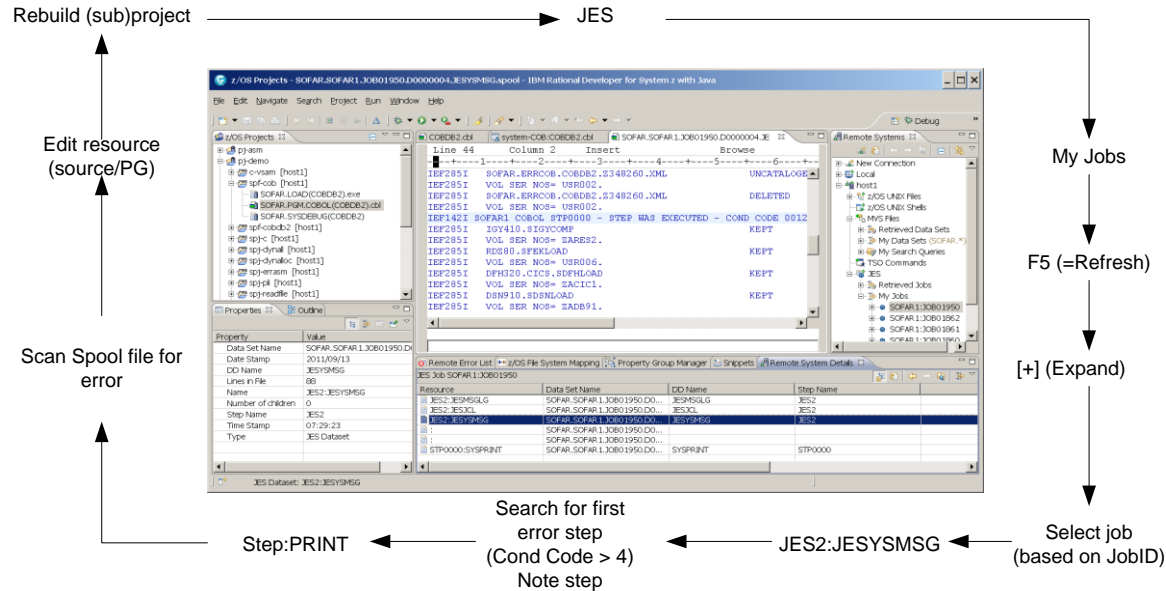
Integrated error processing

RDz provides more information should the job fail with a JCL error.
This means RDz analyses the JESYSMSG file (it could do more but, unfortunately, does not).



RDz-based development

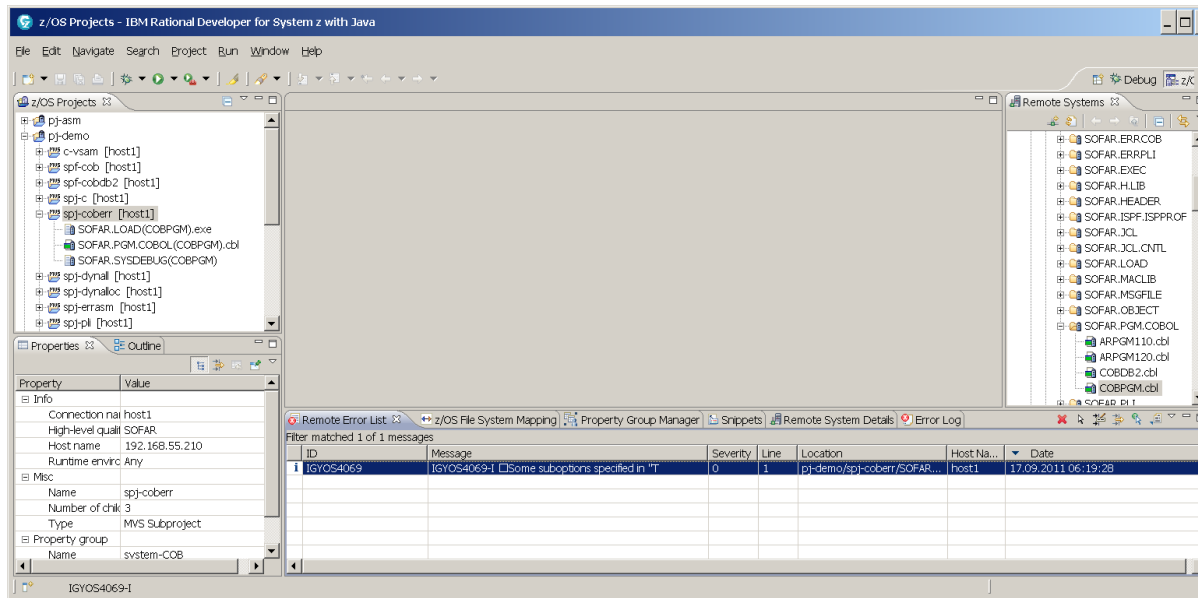
Errors in a non-compiler step processing steps



RDz-based development

Errors in a non-compiler step summary

At the completion of the job, the Remote Error List will contain any compiler messages (possibly none).

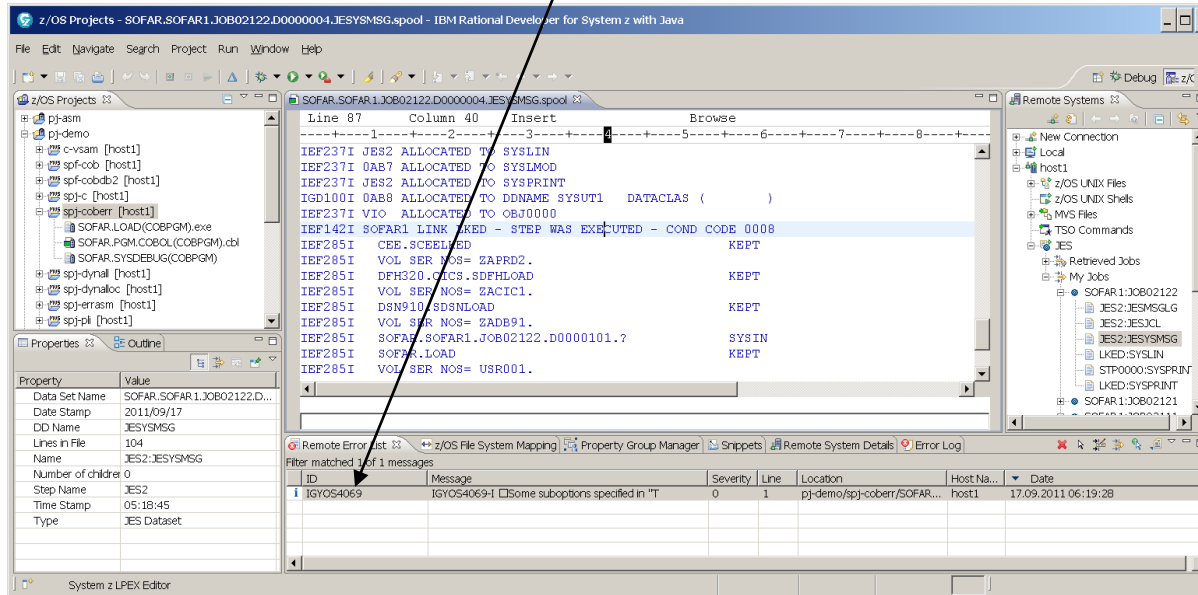


RDz-based development

Processing steps 1

- Step 1:** Rebuild sub(project); note the job number
- Step 2:** Wait for job completion (indicated only indirectly when the "operation in progress" message disappears)
- Step 3:** Refresh My Jobs (F5)
- Step 4:** Search for the required job (using the noted job number from step 1)
- Step 5:** Double-click JESYSMSG to display the job message log
- Step 6:** Browse JESYSMSG spool file and search for the first step completed message with error (cond code > 4)
- Step 7:** Note the associated step name, here LKED

Display of a found step completion message



The screenshot displays the IBM Rational Developer for System z with Java interface. The main window shows a JESYSMSG spool file with the following content:

```
Line 87 Column 40 Insert Browse
-----1-----2-----3-----4-----5-----6-----7-----8-----
IEF237I JES2 ALLOCATED TO SYSLIN
IEF237I 0A87 ALLOCATED TO SYSLOAD
IEF237I JES2 ALLOCATED TO SYSPRINT
IGD100I 0A88 ALLOCATED TO DDNAME SYSUT1 DATACLAS ( )
IEF237I VIO ALLOCATED TO OBJ0000
IEF142I SOFAR1 LINK LKED - STEP WAS EXECUTED - COND CODE 0008
IEF285I CEE.SCEBLDED KEPT
IEF285I VOL SER NOS= ZAPRD2.
IEF285I DFH320.CICS.SDFHLOAD KEPT
IEF285I VOL SER NOS= ZACIC1.
IEF285I DSN910.SDSNLOAD KEPT
IEF285I VOL SER NOS= ZADB91.
IEF285I SOFAR.SOFAR1.JOB02122.D0000101.7 SYSIN
IEF285I SOFAR.LOAD KEPT
IEF285I VOL SER NOS= USR001.
```

The error message is highlighted in blue. Below the spool file, the Remote Error List shows the following error:

ID	Message	Severity	Line	Location	Host Na...	Date
1	IGY054069 IGY054069-I Some suboptions specified in "T	0	1	pj-demo/spj-coberr/SOFAR...	host1	17.09.2011 06:19:28

The Properties window on the left shows the following details for the selected error:

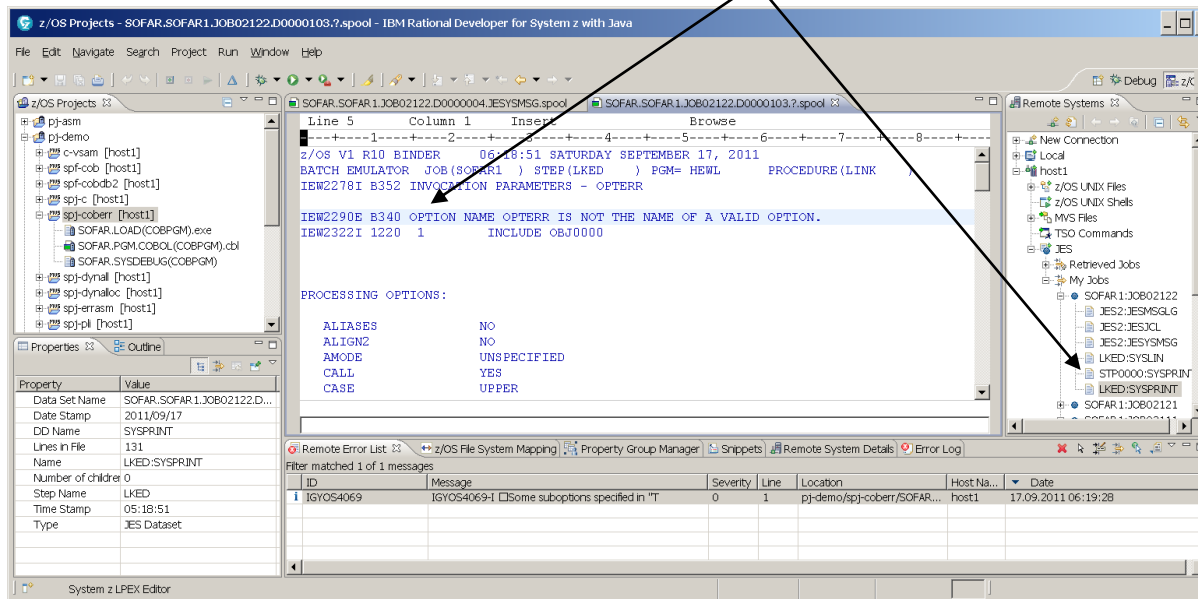
Property	Value
Data Set Name	SOFAR.SOFAR1.JOB02122.D...
Date Stamp	2011/09/17
DD Name	JESYSMSG
Lines in File	104
Name	JES2:JESYSMSG
Number of childnr	0
Step Name	JES2
Time Stamp	05:18:45
Type	JES Dataset

RDz-based development

Processing steps 2

Step 8: Double-click the step print file

Step 9: Browse step print file - search for error messages



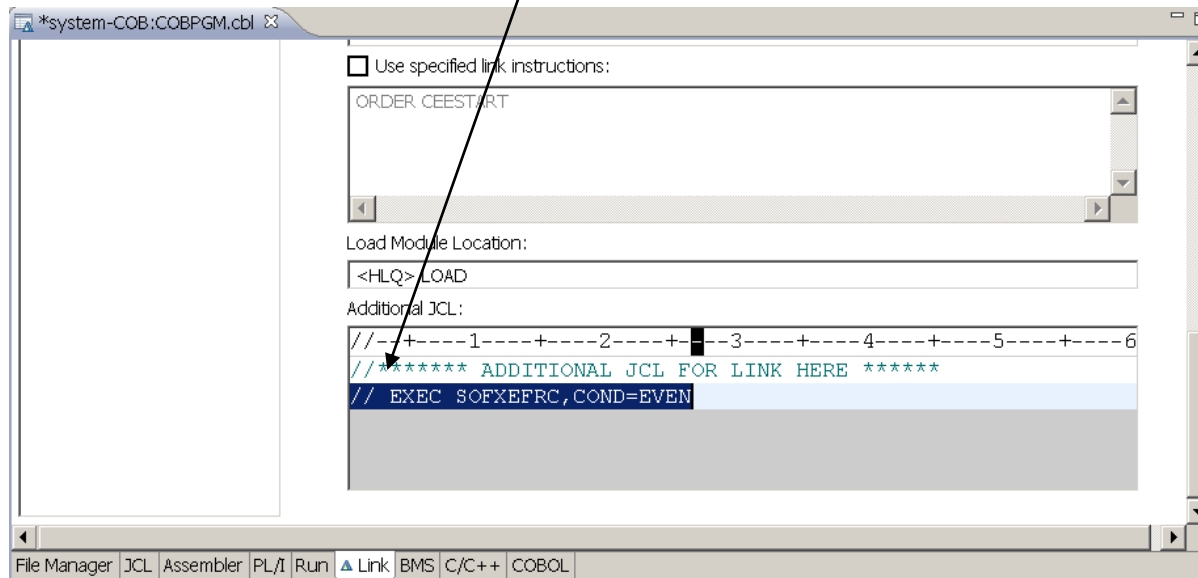
Step 10: Correct the appropriate resource (source program or property group) depending on the error

RDz-based development

The first better solution 1

Error Feedback Step Return Code (SOFXEFRC procedure)
Specify **SOFXEFRC** procedure as Additional JCL in the last step

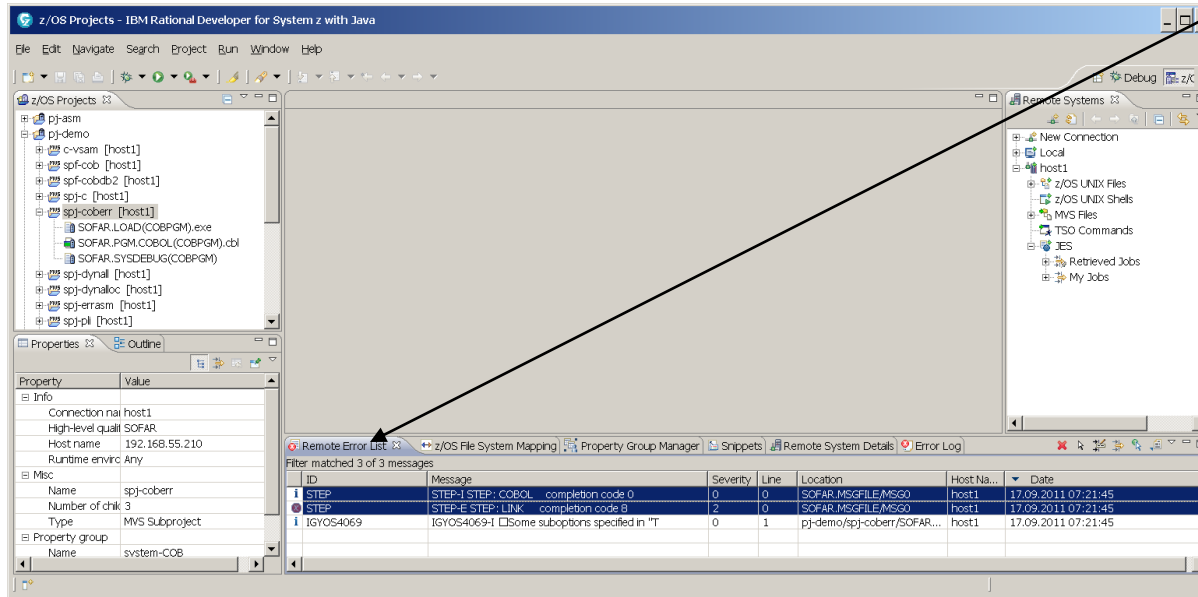
```
// EXEC SOFXEFRC ,COND=EVEN
```



RDz-based development

The first better solution 2

Step condition codes are added to the standard error feedback messages in the Remote Error List



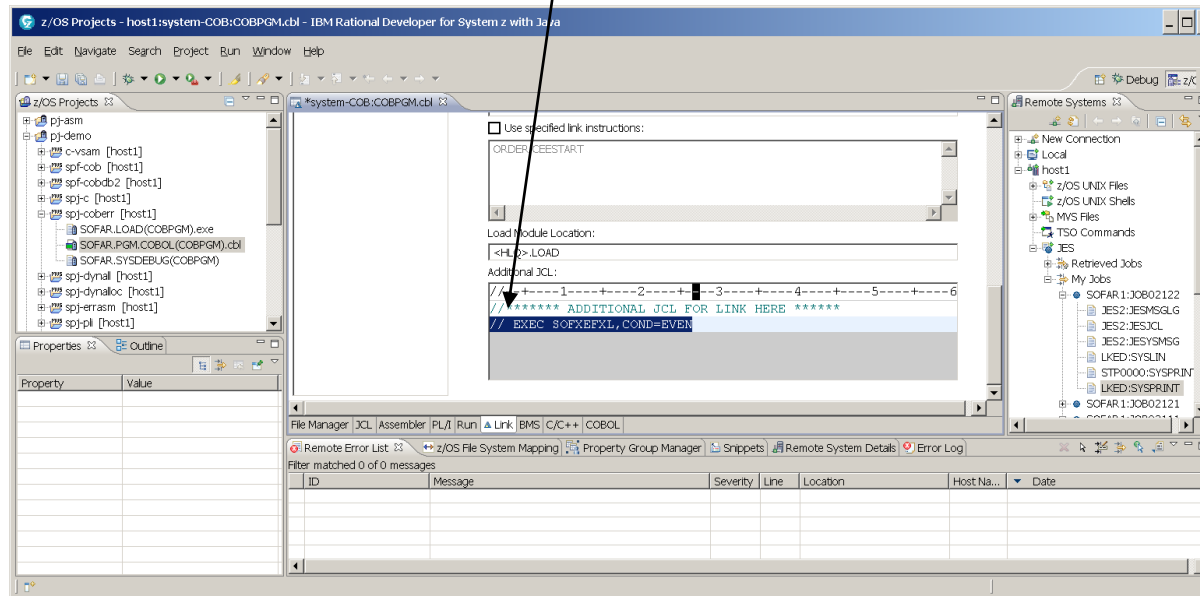
This shows immediately which steps (if any) failed and any steps skipped.
SOFXEFRC provides positive feedback procedure.

RDz-based development

The even better solution 1

Error Feedback eXtra Large (SOFXEFXL Procedure)
Specify **SOFXEFXL** procedure as Additional JCL in the last step

```
// EXEC SOFXEFXL,COND=EVEN
```



RDz-based development

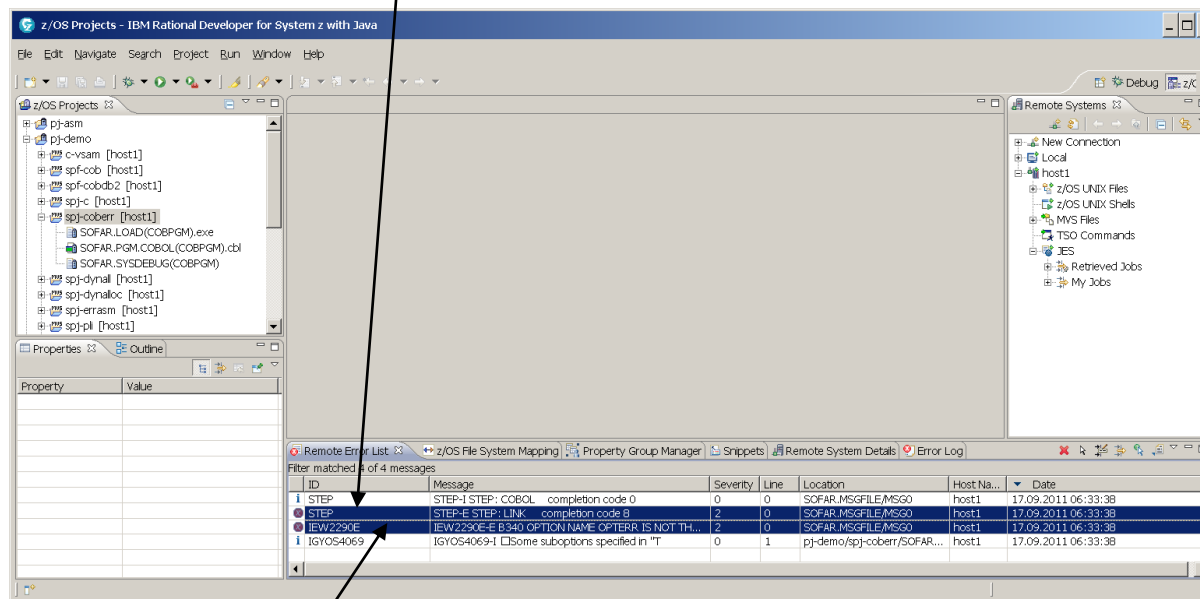
The even better solution 2



SOFORTE
SOLUTIONS FOR TEAMS

Direct display in the Remote Error List of the job steps (with step completion code) and any error messages

Step completion code



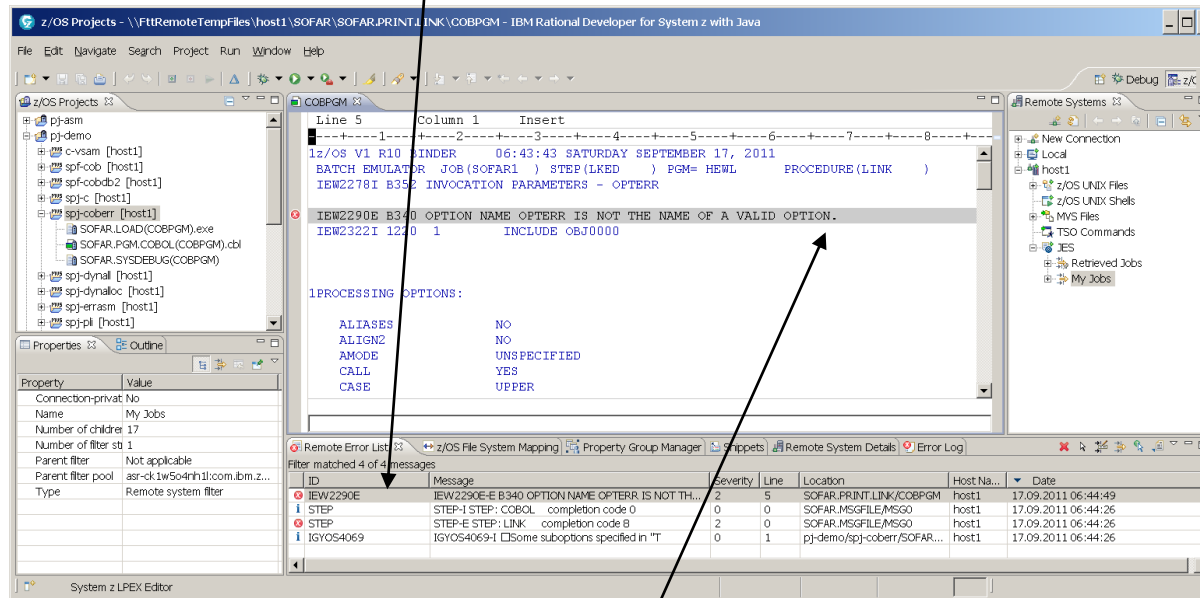
Error message

RDz-based development

The even better solution 3

If a non-spool print file is specified, a double-click on the message in the Remote Error List displays the associated print file with the cursor positioned at the error.

Double-click error message



Error message in the associated print file

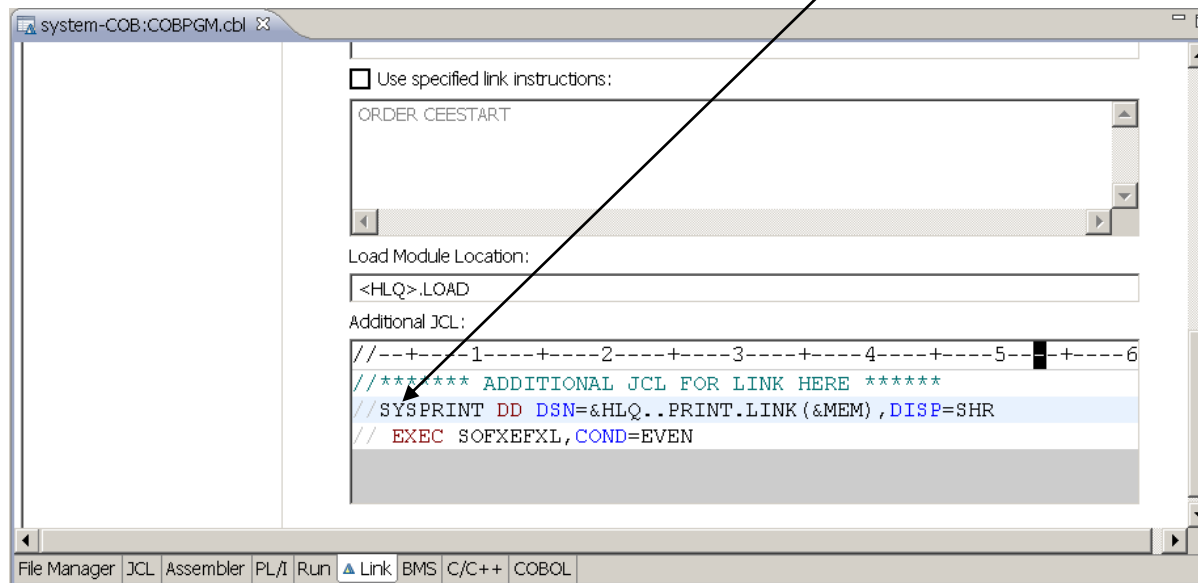
RDz-based development

The even better solution 4

How to specify a non-spool print file

Associated Additional JCL (for example):

```
//SYSPRINT DD DSN=&HLQ..PRINT.LINK (&MEM) ,DISP=SHR
```

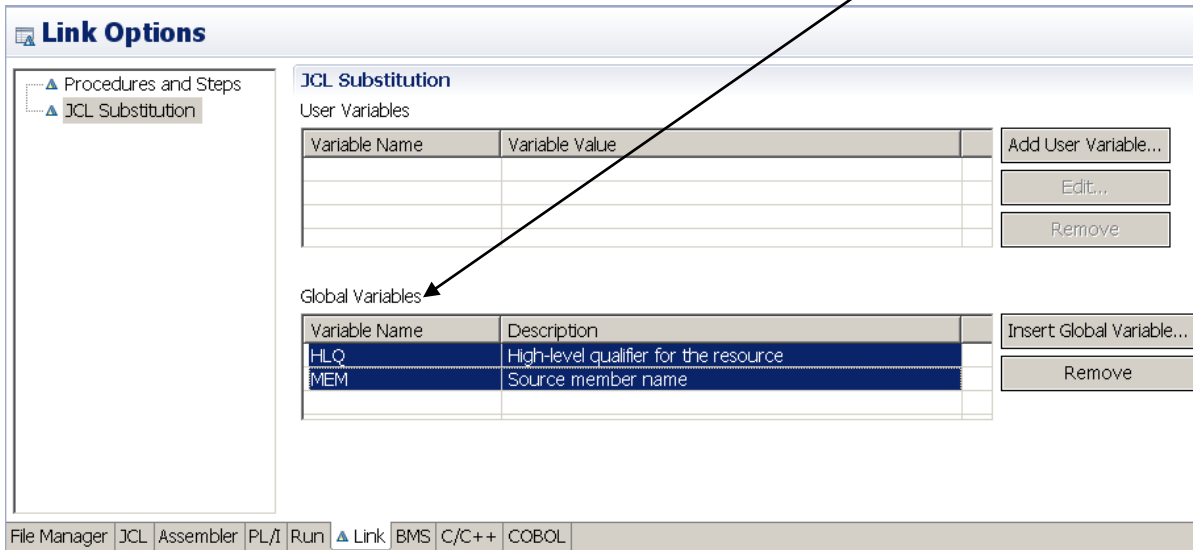


(RDz) Restriction: The print dataset must be a PO file. In this case, the &HLQ and &MEM variables are set as Global Variables in the JCL Substitution:

RDz-based development

The even better solution 5

Where possible define automatic (global) variables



Link Options

▲ Procedures and Steps
▲ JCL Substitution

JCL Substitution

User Variables

Variable Name	Variable Value

Add User Variable...
Edit...
Remove

Global Variables

Variable Name	Description
HLQ	High-level qualifier for the resource
MEM	Source member name

Insert Global Variable...
Remove

File Manager JCL Assembler PL/I Run ▲ Link BMS C/C++ COBOL

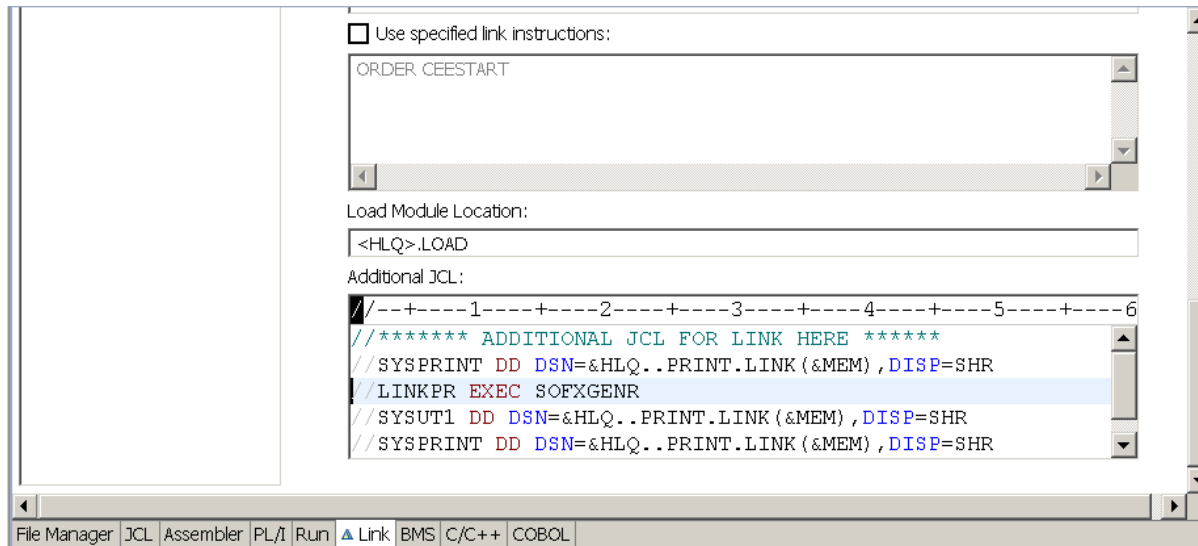
The advantages of Global Variables is that they are set automatically for the user (HLQ) and program (MEM).

RDz-based development

The even better solution 6

How to print a non-pool file

An additional step can be used to route a non-pool file to SYSOUT, such as with IEBGENER, here invoked with the SOFXGENR procedure).

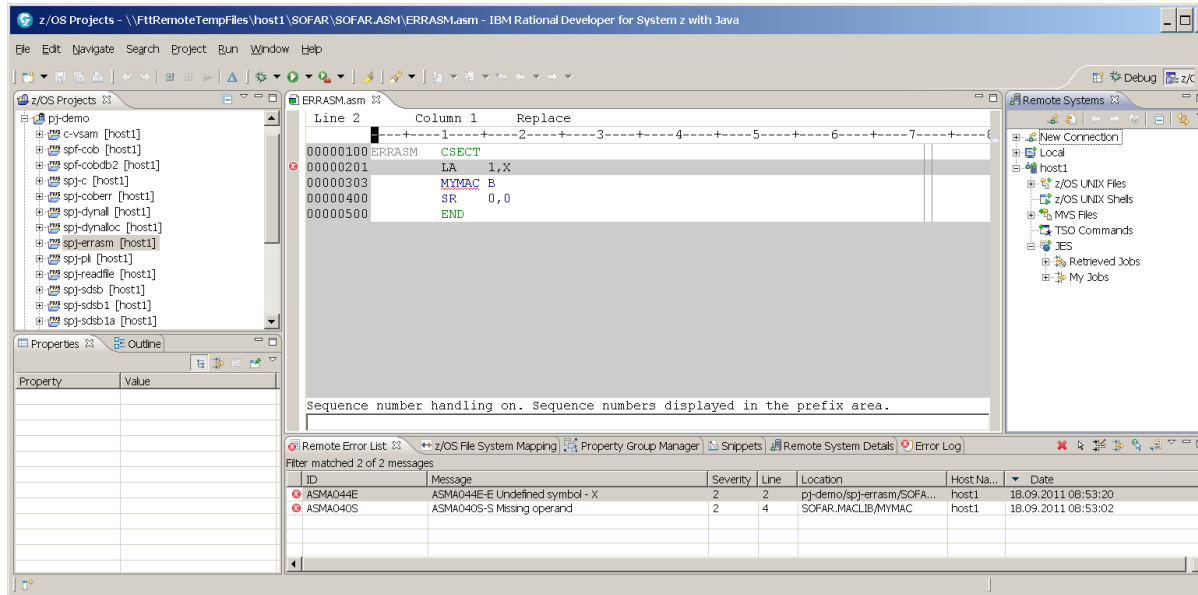


RDz-based development

Other error feedback considerations - Assembler

RDz currently (RDz 8.0), as standard, offers only limited error feedback support for Assembler. Namely, generate a SYSXMLSD file if the Support Error Feedback checkbox is activated; the user is responsible for populating this file (for example, with an appropriate Assembler exit or postprocessor).

The following screenshot shows the SoforTe error feedback implementation for Assembler using an exit (SOFXMGUA).



SOFXMGUA behaves in a similar manner as the standard RDz error feedback provided for COBOL.

RDz-based development

Other error feedback considerations - Assembler

The SoforTe Assembler error feedback requires the following Assembler settings

Assembler Options:

ADATA,EXIT(ADEXIT(SOFXMGUA))

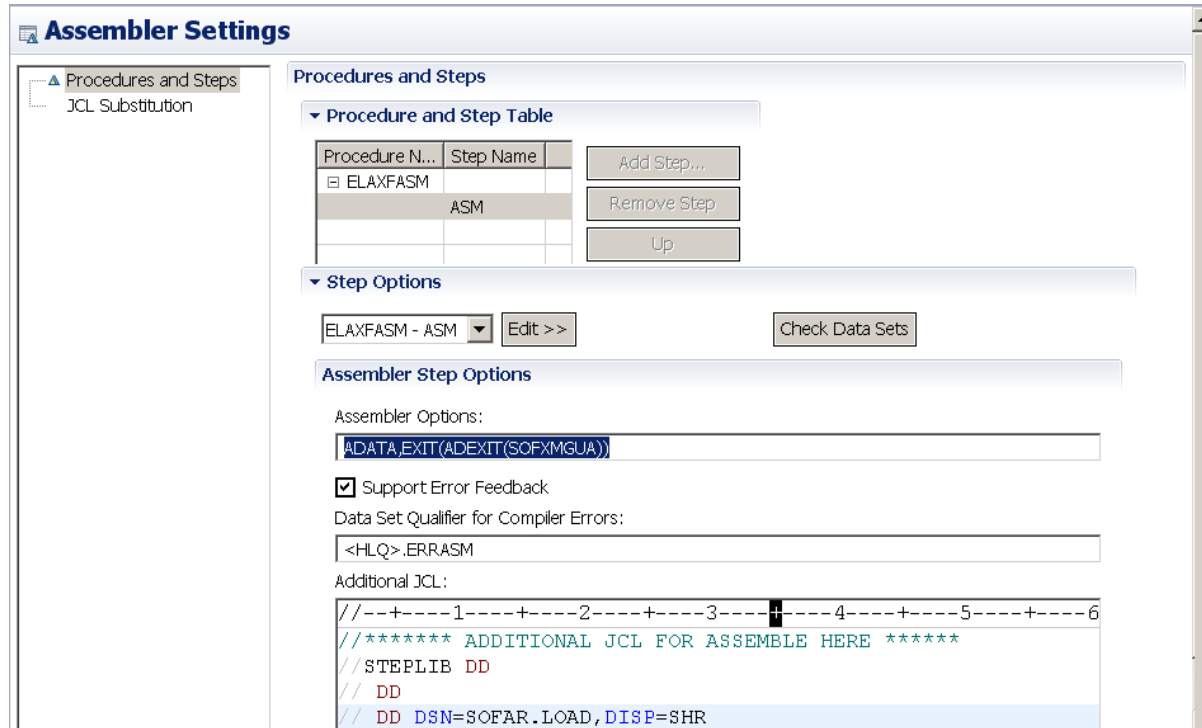
Add the load library to the Assembler procedure (ELAXFASM or a customised version) or add the appropriate STEPLIB as Additional JCL, for example

```
//STEPLIB DD  
// DD  
// DD DSN=SO FAR . LOAD , DISP=SHR
```

RDz-based development

Other error feedback considerations - Assembler

Example Assembler Settings customised for Error Feedback

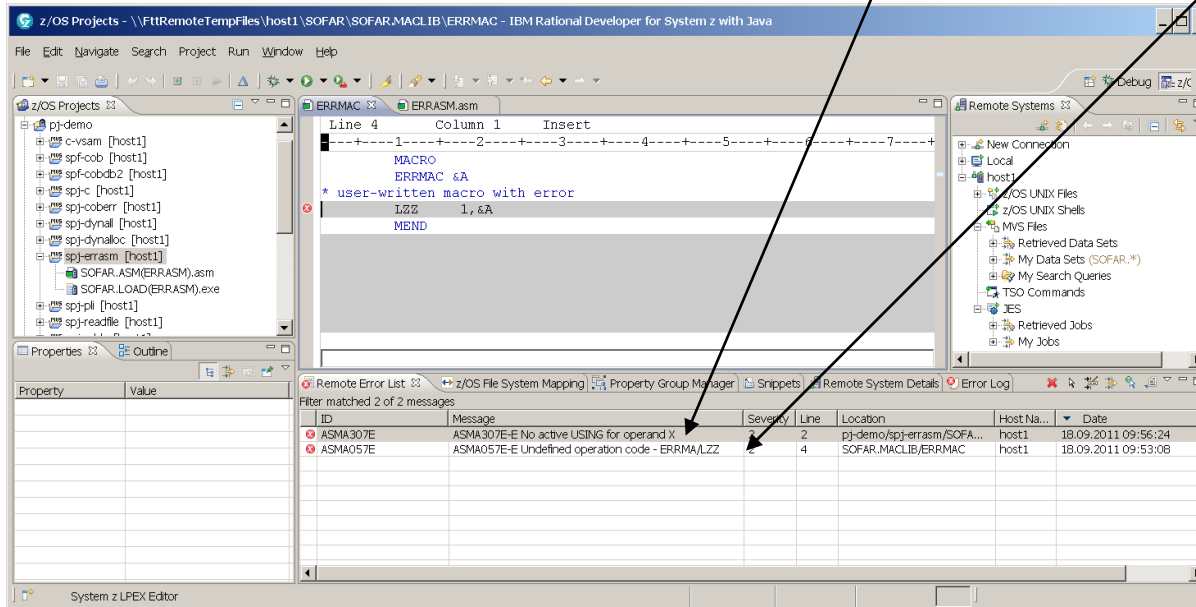


Note: This screenshot has been changed to remove inappropriate settings. The load library (here SOFAR.LOAD) is the library that contains the Assembler Error Feedback exit module (SOFXMGUA).

RDz-based development

Other error feedback considerations - Assembler

Example error feedback output with errors in the primary source and in an invoked macro.



The screenshot shows the IBM Rational Developer for System z interface. The main editor window displays the source code for a macro named ERRMAC.asm. The code is as follows:

```
Line 4      Column 1      Insert
-----1-----2-----3-----4-----5-----6-----7-----+
MACRO
  ERRMAC &A
* user-written macro with error
  LZZ      1, &A
MEND
```

The Remote Error List at the bottom shows two messages:

ID	Message	Severity	Line	Location	Host Na...	Date
ASMA307E	ASMA307E-E No active USING for operand X	Warning	2	pj-demo/spj-errasm/SOFA...	host1	18.09.2011 09:56:24
ASMA057E	ASMA057E-E Undefined operation code - ERRMA/LZZ	Error	4	SOFAR.MACLIB/ERRMAC	host1	18.09.2011 09:53:08

Two arrows point from the text above to the error messages in the Remote Error List. One arrow points to the ASMA307E message, and the other points to the ASMA057E message.

RDz-based development

Other error feedback considerations - other "building sites"



SOFORTE
SOLUTIONS FOR TEAMS

With the exception of the Assembler error feedback exit, all the solutions discussed in this session involve postprocessors.

When preprocessors are used (such as the IBM DB2 precompiler, the CICS translator, and user-written preprocessors), the original source code is changed and normally new code added. This means that the source code used as input to the compiler is not the original source and so the line numbers for error messages do not correspond to the original source. Consequently, unless measures are adopted, any error messages displayed in the Remote Error List are only of limited use, RDz provides capabilities for the error feedback preprocessor implemented with SYSXMLSD to provide the appropriate displacements for the generated code. This approach requires that the associated preprocessor is customised appropriately to produce the required SYSXMLSD file. In most cases, however, such customisation is not available.

Another approach is to write a postprocessor (or compiler exit) that corrects the line number in the error feedback record to reflect the original source.

RDz-based development

Error feedback as implemented in RDz



Depending on the programming language, RDz uses two methods for implementing error feedback:

- XML file (for COBOL, PLI)
- Event file (for C/C++)

The format of the XML file is described in the RDz help. The format of the event file is described in the C/C++ User's Guide.

RDz-based development

Error feedback as implemented in RDz (continued)

Error feedback files:

- SYSXMLSD (XML file, for COBOL, PLI, (Assembler))
 - Preallocated by RDz
 - This requires use of the TSO Command Server
- SYSEVENT (Event file, for C/C++)
 - Allocated by JCL

RDz-based development

Error Feedback - EVENT records

The C/C++ EVENTS compiler option generates EVENT records in the DD:SYSEVENT data set.

The dataset name of the SYSEVENT data set is defined as EVENT FILE DATA SET (C/C++ Settings properties).

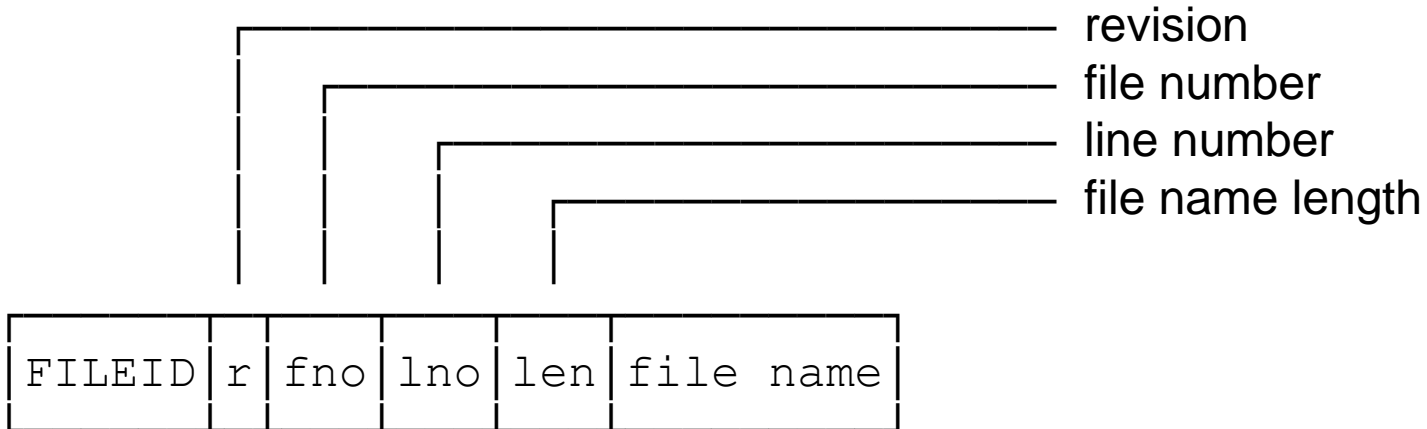
Provided this file exists and is populated (EVENT compiler option), RDz processes the SYSEVENT data set to populate the Remote Error List.

There are three SYSEVENT record types:

- File identifier (FILEID)
- File end (FILEEND)
- Error (ERROR)

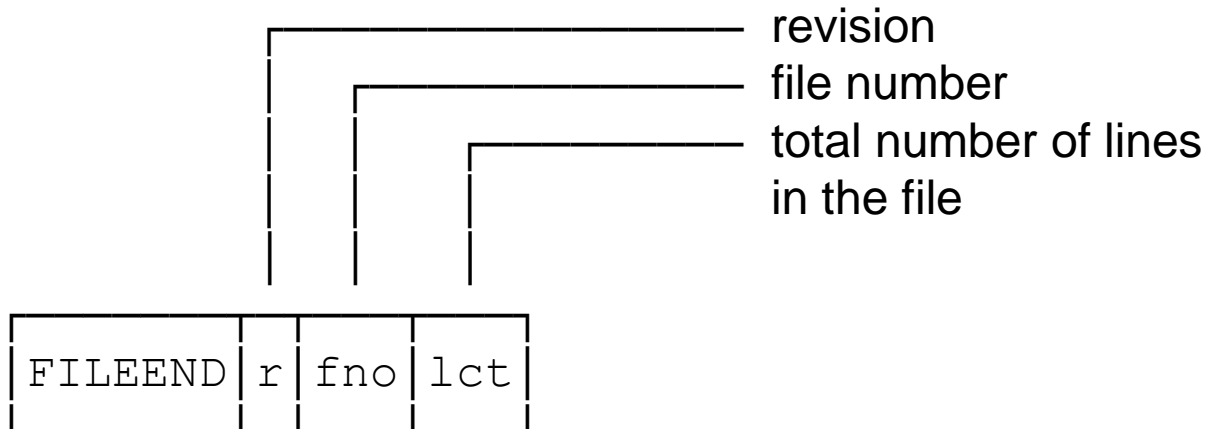
RDz-based development

Error Feedback - File identifier EVENT record



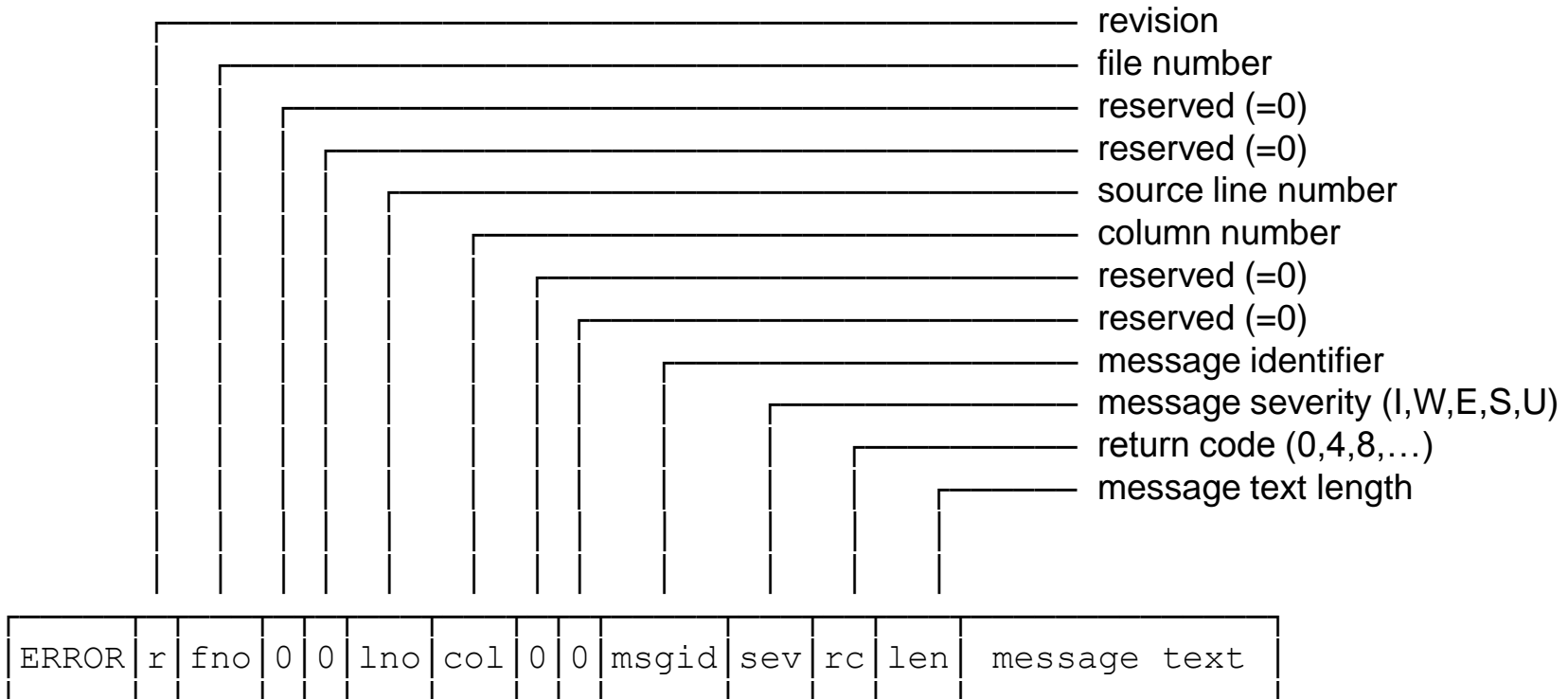
RDz-based development

Error Feedback - File end EVENT record



RDz-based development

Error Feedback – Error EVENT record



Note: Each field is separated with a blank.

RDz-based development

Example EVENT file

```
FILEID 0 1 0 22 'SOFAR.C.PGM(ARFUN01) '  
FILEID 0 2 9 19 CEE.SCEEH.H (STRING)  
FILEID 0 3 38 21 CEE.SCEEH.H (FEATURES)  
FILEEND 0 3 1960  
FILEEND 0 2 384  
FILEID 0 4 10 18 CEE.SCEEH.H (STDIO)  
FILEID 0 5 39 22 CEE.SCEEH.SYS.H (TYPES)  
FILEEND 0 5 567  
FILEEND 0 4 1705  
ERROR 0 1 1 0 109 13 109 19 CCN3068 W 4 56 Operation between  
types "int" and "int*" is not allowed.  
ERROR 0 1 1 0 114 5 114 47 CCN3068 W 4 56 Operation between  
types "int*" and "int" is not allowed.  
FILEEND 0 1 189
```

RDz-based development

SYSXMLSD Error Feedback - Tags 1

The SYSXMLSD is *based* on XML and has the following tags.

BUILD	The root tag.
PACKAGE	An alternative root tag; this may be encompassed by the BUILD tag.
FILEREFERENCETABLE	The encompassing element for the tags that specify the files used during the processing.
FILECOUNT	The number of FILE elements in FILEREFERENCETABLE.
FILE	A FILE element exists for each individual file for which messages are produced.
FILENUMBER	The cardinal number of the associated input file (FILENAME).
FILENAME	The name of the file that corresponds to the FILENUMBER tag.

RDz-based development

SYSXMLSD Error Feedback - Tags 2

MESSAGE	A MESSAGE element exists for each individual message.
MSGNUMBER	The message identifier produced by the processor.
MSGLINE	The line in the input file that caused the processor to produce the message.
MSGFILE	The FILENUMBER of the associated input file for which the message was produced.
MSGTEXT	The text of the message associated with MSGNUMBER.

RDz-based development

SYXMLSD Error Feedback - Tags 3

The following tags are used as preprocessor output.

OUTFILEREFERENCETABLE

This tag describes the files produced by the processor.

OUTFILENUMBER

This tag describes the file number of an individual file produced by the processor.

OUTFILENAME

This tag describes the file name of an individual file produced by the processor.

STATEMENTTABLE

This tag describes the statements that are produced during a process. The description is an ordered quad. Each entry is read as follows: (A,B,C,D), where:

A = Line number written in the output file.

B = File number of the output file.

C = Line number of the input file that was read to generate line number A in file number B.

D = File number of the input file which contained line number C which was read to generate line number A in file number B.

RDz-based development

SYSXMLSD Error Feedback - SYSXMLSD example, part 1

```
<PACKAGE>
  <FILEREFERENCETABLE>
    <FILECOUNT>2</FILECOUNT>
    <FILE>
      <FILENUMBER>1</FILENUMBER>
      <FILENAME>SOFAR.ASM(ERRASM)</FILENAME>
    </FILE>
    <FILE>
      <FILENUMBER>2</FILENUMBER>
      <FILENAME>SOFAR.MACLIB(ERRMAC)</FILENAME>
    </FILE>
  </FILEREFERENCETABLE>
```

RDz-based development

SYSXMLSD Error Feedback - SYSXMLSD example, part 2

```
<MSGTABLE>
  <MESSAGE>
    <MSGNUMBER>ASMA307E-E</MSGNUMBER>
    <MSGLINE>2</MSGLINE>
    <MSGFILE>1</MSGFILE>
    <MSGTEXT>No active USING for operand X</MSGTEXT>
  </MESSAGE>
  <MESSAGE>
    <MSGNUMBER>ASMA057E-E</MSGNUMBER>
    <MSGLINE>4</MSGLINE>
    <MSGFILE>2</MSGFILE>
    <MSGTEXT>Undefined operation code - ERRMA/LZZ</MSGTEXT>
  </MESSAGE>
</MSGTABLE>
</PACKAGE>
```

RDz-based development

Error Feedback - SoforTe extensions



SOFORTE
SOLUTIONS FOR TEAMS

SOFXEFRFC (JCL procedure)

Add step completion codes to the Remote Error List

ERXJES('GETSTEPCC') - retrieve step completion codes

ERXJES('GETFIRSTDDN',ddn) - retrieve DSN for SYSXMLSD, SYSEVENT

ARPGM12 - COBOL program to populate SYSXMLSD file

SOFXEFXL (JCL procedure)

Add step completion codes to the Remote Error List

Add selected postprocessor (LinkEditor, DB2 Bind) error messages
to the Remote Error List

ERXJES('GETSTEPCC') - retrieve step completion codes

ERXJES('GETFIRSTDDN',ddn) - retrieve DSN for SYSXMLSD, SYSEVENT

ERXJES('GETSTEPDDN',ddn,sname) - retrieve DSN

ERXJES('BROWSESPPOOL',dsn) - browse spool file

ARPGM12 - COBOL program to populate SYSXMLSD file

SOFXMGUA (Assembler ADATA exit, SPC program)

getFirstDDN() - C function to retrieve DD:SYSXMLSD

ARPGM11 - COBOL subprogram to populate SYSXMLSD file

SoforTe – Solutions for Teams



Thank you for your attention