		_			
		_	-	<u> </u>	
				_	
_	_	_	_	=	_
_	_	_	_	-	

# **Remote Development**

Development of a remote applications using Rational Developer for System z V7.5.

Lab Version V4.02

Last Updated: Monday, 10 August, 2009

# Overview

This lab will show you how to develop applications running on a System z Mainframe system. You will define a remote z/OS connection, set up a MVS project, remote edit, compile and debug.

The lab can be divided into the following main tasks:

- 1. Initial preparation for RDz remote development
- 2. Working with remote programs
- 3. Compiling/linking/executing the remote program

### Contents

Remote Deve	lopment	1
Overview		2
Lab Details		3
1 Initial pre	paration for RDz remote development	5
1.1 Defi	ne and connect to a remote system	6
1.2 Wor	k with MVS files	13
1.3 Adju	ist the editor	18
1.4 Use	JES (Job Entry System)	21
1.4.1	Explore some editor options	21
1.4.2	Submit JCL and review output	
1.5 Expl	ore USS	
1.6 Opti	onal - Explore TSO Commands	31
2 Work with	n remote files	35
2.1 Retr	ieve and import property group	36
2.2 Crea	ate an remote project	
3 COBOL.		41
3.1 Veri	fy project properties	42
3.2 Crea	ate a COBOL HelloWorld program using zAPG	51
3.3 Add	resources to your project	54
3.4 Expl	oring the Editor	55
3.5 COE	BOL Syntax Checking	57
3.6 Com	pile/link/execute the remote COBOL program	61
3.6.1	Using Project Build	61
3.6.2	Using JCL Generation	61
3.6.3	Using Menu Manager	63
3.7 Deb	ugging Cobol	64
3.7.1	Some more editor goodies	64
3.7.2	Modified property group for the JCL generator	
3.7.3	Right Click Menu	68
374	l Iser Exits	68
375	Using Menu Manager	88
376	Debug	00
0.7.0 20 Con	Ubuly	
3.8 Cop	ybook expansion and dependency check	77

#### Lab Details

Description	Value	Comment		
	Local			
RDz Install path	C:\Program Files\IBM\SDP70			
Appserver Path	C:\Program			
	Files\IBM\WebSphere\AppServer			
	Host general			
Userid	USER##	TSO user		
Password	javaws			
Hostname / IP	192.168.7.xx	xx varies from 71 to 84		
JES Port	9715	JES Job Monitor		
MVS / USS Port	4039	Remote RSE Daemon		
Remote Connection Profile	HostProfile			
User's home directory	/u/user##	USS home directory		
User's HLQ	USER##			
Prepared Workshop Files				

Dataset pre	fix for prepared	AR	NOLD	
		//		
workshop fi	lies	/100	al/zoaws	
	A	Npp L	Development Paths and Libs	
Java home	directories	/usi	r/lpp/java/J1.4	Java 1.4.2 SR6 31 Bit
		/usi	r/lpp/java/J1.5	Java 1.5.0 SR2 31 Bit
Link Librarie	es for COBOL	SY	S1.SCEELKED	
Compile				
Debug Libra	ary HLQ	SY	S1.DEBUG.V710	
Developer f	or z Library	SY	S1.RD4Z.V750	
<b>I</b>			CICS Properties	•
CICS USS i	nstall directory	/usi	r/lpp/cicsts/cicsts32	
CICS user of	directory	/u/c	cicsts/cicsts32	
CICS var di	rectory	/var/cicsts/		
CICS prefix		cics	sts32	
CICS sit		CIC	STS32.CICS1.SYSIN(CICS1)	
CICS Load	Library	CIC	STS32.CICS1.SAMPLE.LOAD	
Additional L	ink Library for	SY	S1.CICS.V320.SDFHLOAD	
CICS Comp	oile			
Catalog Ma	nager	SY	S1.CICS.V320.SDFHSAMP	
CICS Applid		CICS1		
CICS Debug port		9876		
CICS Web Service port		360	02	
CICS ADM port		81		
			Property Groups	
	Batch	_	CICS	IMS
COBOL batchCobol75			CICSCobol75	

	Batch	CICS	IMS
COBOL	batchCobol75	CICSCobol75	
PLI			
C/ C++			
Assembler			

General Hints: Everybody will get a different userid, distinguished by numbers. In your lab guide this individual number is represented by ##. Please always substitute ## by your number.

# 1 Initial preparation for RDz remote development

Before we can start programming some preparations have to be done.

First you will connect to a remote system which has been set up for you. You will learn how to organize your data using filters and how to allocate data sets using RDz. You will also use the workbench to submit JCL which will setup all required data for this workshop.

## 1.1 Define and connect to a remote system

The RSE's Remote Systems view shows all existing connections to remote systems. Connections are System Connection objects that are persisted, containing the information needed to access a particular remote host. The view contains a prompt to create new connections, and pop-up menu actions to rename, copy, delete, and reorder existing connections.

Connections contain attributes, or data, that is saved between sessions of the workbench. These attributes are the connection name, the remote system's host name and system type, an optional description, and a user Id that is used by default by each subordinate subsystem, at connection time. Underneath, all connections are stored as files in an Eclipse project named RemoteSystemsConnections, which the user can enable for team support, allowing connections to be shared by a team.

1. If Rational Developer for System z (RDz) is not already up and running, start it. When prompted for a workspace location you can either continue using a workspace from a previous lab or create a new workspace.

O Workspace Launcher		×
Select a workspace		
IBM Rational Developer for System z stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.		
Workspace: C:\Documents and Settings\Administrator\IBM\rationalsdp7.5\workspace	•	Browse
Use this as the default and do not ask again		Cancel

- 2. If you created a new workspace a welcome screen will be displayed that you will have to close.
- 3. Ensure you are working with the z/OS perspective by selecting Window → Open Perspective → Other... and select z/OS Projects.
- 4. Click on the Remote Systems View and expand the New Connection node, right-click on z/OS... and select New Connection to open the pop-up menu.



5. If this is the first time that you have attempted to create a connection in RSE, you are prompted to create a profile before you can create the new connection. Name your profile HostProfile and Click Next.

If the screen does not appear there are already existing profiles and you can ignore this step.

💿 New	×
Name personal profile	
Uniquely name user profile	~
Welcome to Remote Systems. Connections can be sharable by the team or private to you. Enter a profile name to uniquely identify you from your team members. You will decide for each new connection whether it is owned by the team profile or your profile.	_
Profile: HostProfile	
Image: Conceleration         Mext >         Einish         Canceleration	

In the Parent profile field, the profile named after your workstation appears by default (Note that after you create the connection, you can share this profile to allow other users to have this connection in their RSE perspective).

6. In the Connection name field, type a unique name to identify your connection in the Remote Systems view called

demomvs.

The label that you assign to this connection will help you to differentiate between multiple connections to the same type of remote system.

In the Host name field, type

192.168.7.xx

as hostname or IP address of the z/OS system that you installed the RSE server on. Optionally in the Description field, provide a short description of the z/OS system that you want to connect to.

To verify that the hostname or IP address in the Host name field is valid, select the Verify host name check box.

New Connection		
Remote z/OS S	ystem Connection	
	nomation	
Parent profile:	L3KXY98-1951CZ1	•
Linet arms.	100 100 7	
Connection name:	demomys	
Description:	connection to demo host	
Verify bost par	e	
je veny nosenan		
?	< Back Next > Einish	Cancel

7. Click Next to continue.

8. In the USS Files window we will also use the Remote daemon. Use 4039 as port. Note that authentication will be via userid/password.

New Connection	
z/OS UNIX Files	-
Define subsystem information	
Indicate how the remote server should be launched by default	
Daemon Port (1-65535) 4039 <u>A</u> uthentication method userid/password	~
○ R <u>E</u> XEC	
Path to installed server on host	
dstore	
Server launch command ./server.zseries Port (1-65535) 51	2
Auto-detect SSL	
Use SSL for network communications	
O Connect to running server	
Use SSL for network communications	
⊖ss <u>H</u>	
Path to installed server on host	
dstore	
Server launch command ,/server.zseries Port 22	
Password authentication	
○ <u>K</u> ey authentication	
?     < <u>Back</u> <u>Einish</u> Ca	ncel

9. For the MVS files, enter **4039** as Remote Daemon port.

New Connection	
IVS Files	
Define subsystem information	
Indicate how the remote server should b	be launched by default
<u>R</u> emote daemon	
Daemon Port (1-65535) 4039	Authentication method userid/password 💙
○ R <u>E</u> XEC	
Path to installed server on host	
dstore	
Server launch command ./server	.zseries Port (1-65535) 512
Auto-detect SSL	
Use SSL for network communicat	tions
O Connect to running server	
Use SSL for network communicat	tions
OSSH	
Path to installed server on host	
dstore	
Server launch command //server	zseries Port 22
() Password authentication	
C Key authentication	
3	< Back Next > Einish Cancel

# 10. Click Next.

Note: If you have errors during the connection creation it is because the z/OS system name is not correct or not available (since you specified verify the host name on the step above).

- 11. In the JES Job Monitor Port field, insert
  - 9715

as the port on which the Remote Job Monitor is listening.

In the "Max Number of Lines to Download" field, type the number of lines to download before prompting you to specify if you want to download all of the lines in the dataset. We will accept the default of 5000.

💽 New Connection	×
JES	
Define subsystem information	
JES Job Monitor Port (1-65535) 9715	
Max Number of Lines to Download (1-2147483647) 5000	ĺ
< Back         Mext >         Einish         Cancel	

- 12. Click Finish to create the new z/OS connection and add it to the RSE perspective.
- 13. If network to the to z/OS is available, you will have the connection created as shown below:



14. Expand demomvs to see the sub categories of your remote system.



- 15. Now connect to your system by right-clicking demomvs and selecting "connect" from the context menu.
- You will be prompted for your z/OS userid and password. Type the assigned userid and password. Click on Save user ID and Save password. You will be automatically connected at later times.

💽 Enter Pass	sword	×
System type: Host name:	z/OS 192.168.7.xx	
User ID:	USER##	
Password:	javaws	
	Save user ID	
	Save password	
	<u>0</u> K	<u>C</u> ancel

Click OK to connect to MVS Files subsystem.

17. Since RDz Version 7.5 you can secure your connection using SSL. If this has not been done you will be informed by the following message dialog. Click Yes.

💽 RSEC2315				×
i)	Connection 192. 168. 7. 226 has not been secured using SSL. Proceed anyway? Do not show this message again	Yes	No	

 If you successfully connect to the remote system, the demomvs icon changes to demomvs (note the little green arrow).

## 1.2 Work with MVS files

At this point you are ready to start working with the z/OS assets. We will see how to allocate new PDS (Partition Data Set), copy members and use filters to display data.

19. Expand MVS files.



20. Click on i My Data Sets to see your data sets.

Don't worry if there are several data sets allocated, we will refresh the system later in this chapter.

- 21. Right-click The MVS Files and select "Allocate PDS" from the context menu.
- 22. Use the drop down list to select your High Level qualifier "USER##" and type the data set name JCL to allocate a data set that will contain our JCL members later on.

💽 New Data Set	
Allocate PDS Allocate a new PDS residing on z/OS.	
Host Short Name:	
demomvs	<b>•</b>
Data Set Name:	
JUSER##	
<         Back         Next >         Einish	Cancel

23. Click Next.

24. Select Specify characteristics by usage. Choose "SOURCE" for category and "JCL" for type and click "Next".

🔘 New Data Set	<u>_ D ×</u>
Data Set Allocation Choose category and/or type.	
Data Set Name: ZUSER10.JCL Copy characteristics from an existing data set:	- Duemon - 1
<ul> <li>Specify characteristics by usage type:</li> <li>Category SOURCE</li> <li>Type JCL</li> <li>Specify characteristics (Advanced allocation).</li> </ul>	browse
? < <u>Back</u> Einish	Cancel

This is a nice feature since it allocates a data set that has the characteristics necessary for holding JCL.

25. Note the default values and click Finish.

💽 New Data Set		
Data Set Characte Specify data set chara	eristics cteristics for the new PDS or sequential file.	
Data Set Name: ZU	SER 10. JCL	
Volume Serial:		
Generic Unit:		
Space Units:	BLOCKS	
Primary Quantity:	200	
Secondary Quantity	: 100	
Directory Blocks:	20	
Record Format:	FB	
Record Length:	80	
Block Size:	0	
Data Set Type:	LIBRARY(PDSE)	
Expiration Date:		
System Mana	ged Storage	
0	< Back Next > Finish	Cancel

26. If space is available in the z/OS, this Data set will be created. If you do not see it, just right-click on MVS Files and select Refresh.



We will now create a filter to access our master data set ARNOLD.JCL and copy a member from there to our own JCL data set USER##.JCL. You already met a MVS filter when you expanded "My Data Sets" earlier in this lab. This is the default filter to display all data sets of the user currently connected to the system.

27. Right-click MVS-Files and select New  $\rightarrow$  Filter from the context menu.

28. Specify "ARNOLD.JCL" as Filter string and click "Next". Be aware that this is a master dataset, not your own. Please use exactly the name stated in the script here.

🗿 New F	ilter		<u>- 🗆 ×</u>
Filter			
Filter Create a new filter Filter string: ARNOLD.JCL			
			 <b></b>
Filter stri	ng:		
ARNOL	D.JCL		
		1	

29. Specify a filter name (e.g. "lab\_materials") and click Finish.

💽 New Filter			
Filter Name the new filter			-
			$\rightarrow$
Filters are saved for easy re-use. Specify a unique n Systems view, and will be expandable.	ame for this filter. T	his name will appear	r in the Remote
Filter name:	la	ab_materials	
Only create filter in this connection			
Select a profile to own the new filter. This determine be placed in the default filter pool for that profile.	; if it is unique to yo	u, or sharable by th	ie team. It will
Select the profile whose default filter pool is to conta	in the new filter:	3KXY98-1951CZ1	<b>T</b>
? <u>&lt; B</u> ac	k <u>N</u> ext >	<u>F</u> inish	Cancel
The filter should appear in your Remote S	ystems View.		
	-		
E:≦ New Connection E: E Local			
É			
⊥ ····································			
terristic My Data Sets (205ER 10. °)			
My Search Queries			
E B JES	l		

Note (you don't have to do this now), that after creation you could specify more filter criteria by right-clicking your filter and selecting Properties. You could create other filters like HLQ.\*, HLQ.\*.COBOL, HLQ.UTIL.\*, HLQ.\*.COB\*, etc... (where HLQ is your high level qualifier) Using this mechanism you can easily structure your workspace by for example displaying only the datasets required for a certain project using multiple filter criteria.

The following picture shows an example for a filter used in a COBOL programming project for a high level qualifier called HLQ:

Properties for cobol_p	roject		
type filter text	Filter Strings		$\leftarrow \star \Rightarrow \star$
Filter Information Filter Strings Refactoring History	Filter name: Parent filter pool:	cobol_project Hostprofile Filter Pool	
	Filter strings: New filter string HLQ.COBOL HLQ.JCL HLQ.COPYLIB	New filter string: Eilter string:	Create <u>R</u> evert
0			OK Cancel

 Expand the filter and navigate to the member RDZALLOC.jcl. Copy this member to your own JCL dataset you created. You can do this by right clicking the member and selecting Copy.

Then right click USER##.JCL and click Paste.

## 1.3 Adjust the editor

The default profile used by the LPEX editor is lpex, which defines the key behavior. The LPEX editor can emulate most of the key behavior of other editors like ISPF and XEDIT.

31. Open the Preferences dialog (Windows → Preferences) and select LPEX Editor from the left. Do not close the window yet.

O Preferences		
	LPEX Editor	$\diamondsuit \bullet \bullet \bullet \bullet \bullet$
Java EE     Java Script     JavaScript     JPA     Jython     LPEX Editor	Editor profile brief emacs epm spf lpex set	
<ul> <li>Image Menu Manager</li> <li>Image MFS Editor</li> <li>Image Modeling</li> <li>Image Plug-in Development</li> <li>Image Remote C/C++</li> <li>Image Remote Synchronization</li> </ul>	vi xedit	
E Remote Systems	•	Reset Restore Defaults Apply
0		OK Cancel

You now are able to enter prefix commands in the prefix area, such as d to delete, m to move, etc... like you mabe used to from ISPF.

#### 32. Creating Tabs for the Editor

Since we are using COBOL is a good idea to create some TABS for easy navigation in the editor.

Expand LPEX Editor, click on Tabs and enter 1 8 12 as Tab stops. Also change the Tab increment to 4 (instead of 8). Each time Tab key is pressed will move the cursor to positions 1, 8 and 12. The increment 4 will also make stops at 14, 16, 20, and so on.

Preferences		
	Tabs	↓ → → ▼
LPEX Editor	Tab stops 1 8 12	
Block	Tab increment 4	
Compare Controls	Display tab characters expanded	
Find Text ⊕-Parsers		
Print		
	_ <u> </u>	
User Commands	<u>R</u> eset	Restore Defaults Apply
0		OK Cancel

33. Go to General  $\rightarrow$  Compare/ Patch and check "Ignore white spaces". We need this feature later for code comparison.

O Preferences		
	Compare/Patch 🤤 🔹 🔿	- <b>-</b>
General     Appearance     Capabilities     Compare/Patch     Content Types     Editors     Error Reporting     Keys     Network Connection:     Perspectives     Search     Search     Security     Startup and Shutdov     Web Browser     Workspace     Agent Controller     Ant     Auto Comment     Backward Compatibility     Bidirectional Developmen     BMS Map Editor     BMS Map Editor	General       Text Compare         Image: Open structure compare automatically         Show structure compare in Outline view when possible         Image: Show additional compare information in the status line         Image: Show additional compare information in the status line         Image: Image: Show additional compare information in the status line         Image: Image: Image: Show additional compare information in the status line         Image: Image: Image: Image: Image: Show additional compare information in the status line         Image: Imag	
?	OK Can	icel

34. Click Apply and then OK to save your changes.

# 1.4 Use JES (Job Entry System)

With RDz you can edit Job Control Language, submit jobs to JES and review the output all in one development environment.

We will now modify and submit JCL to z/OS to allocate some datasets that are required for this lab.

### 1.4.1 Explore some editor options

We will first use your JCL source to demonstrate some editor features. Those features apply to other sources than JCL as well, use them later if you find them appropriate.

- 35. Double click on the copied RDZALLOC.jcl to open it.
- 36. You can use the mouse to expand the editor area to see more lines of the COBOL program. Also remember that when double clicking in the title (RDZALLOC.jcl) you can either make a full screen or return to original size. Also Window → Reset Perspective returns to the default.
- 37. Change xxxxx to your z/OS userid (USER##) using Ctrl + F and clicking Replace all.

Ð	RDZALLOC.jd 🛛	
	Line 1 Column 8 Insert	
	//+	+
	000001// <mark>xxxxx</mark> JOB ,	
	000002 // MSGCLASS=H, TIME=(,4), REGION=28M, COND=(16, LT)	
	000003 //*	
	000004 //* SETUP JOB FOR WDz v6.01 POT	
	000005//*	
	000006//* NEEDED FOR THE Z/OS LAB EXERCISES	
	000007//* CHANGE 'XXXXX' TO YOUR MVS USER ID	
	000008//* ******************************	
	000009//~ SIEP: DELISI	ΨI
		نے ا
	Find XXXXX Next Previous	All
		<u>-</u>
	Replace USER##	all
	I Case sensitive I Whole word I Regular expression I Wrap I Select found text Pee	k: [(
	Restrict search to selection Restrict search to columns Start column 1 End column	80

38. Hit Ctrl + L and enter 2 to navigate to line number 2.



39. Add a JCL error, for example if you delete the first comma in this line and press enter this will have the error below.

Line 1	Colum	un 1	Insert	1 c	hange				
//	+1	+2	+	-3	-+	-4	-+	-5	-+-
000001 / / US	ER## )B	,							
000002 // 1	SGCLAS <mark>S</mark> =3	HTIME=(,	4), REGIO	N=28M	I, COND	=(16,	LT)		
000002 Unes	pected cl	haracter	"=".						
000003//*									
000004 //*	SETUP J	OB FOR W	Dz v6.01	POT					
									N I

40. Fix the line.

Either type the comma or hit Ctrl+Z to undo your change.

- 41. Save the changes (Ctrl + S). Note that the \* that is beside of the title goes away.
- 42. Have a look at the Outline View where the structure of your code is displayed. You can click within that view to navigate through your code.



43. You could have a situation where you need to see hexadecimal contents of the fields (Common when doing assembler). Right-click and select Source → Hex edit line. You will have the result below:



- 44. Right-Click in any place in the editor area and select View → Open new view. This is similar to the ISPF split screen. Sometimes this can be useful. Remember that double clicking in the title gives you a full screen and much more work space.
- 45. Right-Click again in any place in the editor area and select View → Horizontal split. This is another option. Also see other options like Next View and Previous view.
- 46. To return to the default view, use Right-Click inside of the second view and select View  $\rightarrow$  Close View

47. One nice feature of RDz is the capability to recover previous versions using the local workstation. This is very useful when you delete and save components on the z/OS where undo is not possible.

Right click within the code and select "Replace with  $\rightarrow$  Local History" from the context menu. It will show all the previous versions (probably you have just one).

Compare	
/RemoteSystemsTempFiles/FttRemoteTempFiles/demomvs/ZUS	ER 10/ZUSER 10. JCL/RDZALLOC. jd
Revision Time	
■ 12/18/08 11:32 AM	
Ⅲ 12/18/08 11:32 AM	
Ⅲ 12/18/08 11:31 AM	
Text Compare	A 🕸 4 🔂
Uvrkspace File	E Local History (12/18/08 11:32 AM)
//ZUSER10 JOB ,	//XXXXX JOB,
<pre>// MSGCLASS=H,TIME=(,4),REGION=28M,CC</pre>	// MSGCLASS=H,TIME=(,4),REGION=28
//*	//*
//* SETUP JOB FOR WDz v6.01 POT	//* SETUP JOB FOR WDz v6.01 POT
//*	//*
<pre>//* NEEDED FOR THE Z/OS LAB EXERCI</pre>	//* NEEDED FOR THE Z/OS LAB EX
//* CHANGE 'ZUSER10' TO YOUR MVS US	//* CHANGE 'ZUSER10' TO YOUR MV
//* ***********************************	//* ***********************************
//* STEP: DELIST	//* STEP: DELIST
//*	//*
//* UPIOD TO NILOCATING THEM IN THE	//* (PDIOD TO ALLOCATING THEN IN
//~ (PRIOR TO ALLOCATING THEM IN THE	//* CFRIOR TO ALLOCATING THEM IN
	<u> </u>
(?)	Replace Cancel

48. As we do not want to undo our changes, press Esc to dismiss the dialog.

## 1.4.2 Submit JCL and review output

49. Submit this job by typing "sub" into the command line

```
RDZALLOC.jd 🔀
   Line 1
                Column 14
        //--+---1--------2--
  000001//;USER## JOB ,
  000002 // MSGCLASS=H, TIME=(,
  000003 //*
  000004 //*
            SETUP JOB FOR WD
  000005 //*
  000006 //* NEEDED FOR THE Z
  000007 //* CHANGE 'USER##
  000008 //* *****************
  000009 //* STEP: DELIST
  000010//*
  000011//* DE-ALLOCATE DATA
  4 |
  sub
```

- 50. Note that you could also right-click the member or within the editor and select "submit" from the context menu.
- 51. If the Job is not submitted and you get the message Job Monitor not connected to system as shown below, It is because your JES subsystem is not connected

5	RDZALLOC.jd 🗙
	Line 8 Column 48 Insert
	//+12+3+4
	000001//USER## JOB ,
	000002 // MSGCLASS=H, TIME=(,4), REGION=28M, COND=(1
	000003 //*
	000004//* SETUP JOB FOR WDz v6.01 POT
	000005 //*
	000006 //* NEEDED FOR THE Z/OS LAB EXERCISES
	000007 //* CHANGE 'USER01' TO YOUR MVS USER ID
	000008 //* *****************************
	•
	The Job Monitor server is not connected.
	sub

52. Use the Remote Systems view, locate the note JES under demomvs and Right-click on JES and select Connect. Try to submit again. 53. If submit succeeded you will see the JOB ID that was created for this execution



- 54. Go back to the Remote Systems view and under JES node (must expand demomvs), right click on the "My Jobs" filter and select Refresh (You can also select the filter and hit the F5 key
- 55. Expand the listing and you will see something like:



56. Double click on first listing and be sure that the job has successfully executed and has the return code is 4, 0 and 0 as shown below: (note could all zeros if you had datasets allocated already). If you receive higher return codes contact your lab instructor.

E	) RDZALLOC.jcl	ј јово:	3203.out	×	<b>.</b>							C	- 8
		JES2	JOB	LO	G		SΥ	S T	ΕM	I S	ΥS	1	<b>_</b>
	JOB03203	THUR	SDAY,	18 D	EC 2	800		-					
	JOB03203	IRR010I	USERII	D ZUS	ER10	15	S AS	SIGN	ED I	ΟТ	HIS J	ΟВ.	
	JOB03203	ICH70001	I ZUSER	R10 (	LAST	ACC	ESS	AT	14:2	4:4	4 ON	THU	1
	JOB03203	\$HASP373	ZUSER1	10 S	TART	ED -	- IN	IT 6		- c	LASS	A -	-
	JOB03203	IEF403I	ZUSER10	) – S	TART	ED -	- TI	ME=1	4.24	.45			
	JOB03203	-											
	JOB03203	-JOBNAME	STEP	IAME	PROC	STER	?	RC	EX	CP	CON	N	
	JOB03203	-ZUSER10	DELIS	ST				04		43	2	2 *	
	JOB03203	-ZUSER10	ALLO	2				00		0		0 *	
	JOB03203	-ZUSER10	COPY					00	2	20	26	4 *	
	JOB03203	IEF404I	ZUSER10	) – E	NDED	) — I	IME	=14.	24.4	6			
	JOB03203	-ZUSER10	ENDEI	). N	AME-						TO	TAI	
	JOB03203	\$HASP395	ZUSER1	LO E	NDED	)							
	ES2 JOB ST	ATISTICS -											
	2008 JOB	EXECUTION	DATE										
	130 CARD	S READ											-
	<b>Ⅰ</b>												

You just submitted a JOB that was executed in the z/OS and allocated some data sets that you will need for this lab.

57. Close the editors that are opened and the JES2 JOB LOG.

To display your submitted jobs in a more "SDSF-like" format, you can do the following:

58. Right-click either the JES icon or any job filter and select "Show in Table" from the context menu.



59. The "Remote System Details" View displays your Jobs and corresponding Information in a table.

📕 Remote System Detai	ils 🗙 z/	OS File Syste	m Mapping	Remote Error List	Problems	Console	Remote She	l Remote
Job Filter My Jobs								
Name	Job ID	Job Name	Job Owner	User return	Return In	ifo Reti	urn Status 🛛 J	lob Class
ZUSER 11: JOB08806	JOB08806	ZUSER 11	ZUSER 11	000	NORMAL	CON	PLETION	4

- 60. You can purge the output of jobs listed. To purge Go to the Remote System view, Right-click on your Job ID and select Purge. Alternatively you can purge the job as well from the Remote Systems Details table.
- 61. Using the Remote Systems view, right click on your "My Data Sets" filter and select Refresh (alternatively select the filter and hit F5).
- 62. Expand My Data Sets clicking on the + and you should now have all the required datasets for the next exercise ready.

JES job filters are used to define the search that is done in JES for jobs. JES job filter search parameters include the following: Job Owner, Job Prefix, Job Output Class, Job Status and Job Class, comparable to SDSF. You can use \* as a wildcard character.

The filter "My Jobs" you just used has been created for you by default. To create a new filter (you don't have to do this now), right-click the JES node and select New  $\rightarrow$  New JES Job Filter...



You can define several criteria to limit the jobs displayed. For a later exercise use the prefix GEN\* as Job Name Prefix, hit Next and give this filter a name like *generated*.

🗿 New Filter				
New JES Job Filter Create a JES Job Filter by	y entring stri	ngs. Use * for	wildcard	3
Job Owner:				
&USERID				
Job Name Prefix:				
GEN <sup>#</sup>				
Job Status:				
*				
Job Class:				
*				
Job Output Class:				
*				
?	: Back	Next >	Finish	Cancel

# 1.5 Explore USS

63. To display the built-in USS File Filters, expand USS Files. The three built-in USS filters are My Home, Home, and Root. Expand My Home to see the contents of your home directory.

Note: The My Home filter displays your home directory. The Home filter displays the directory containing your home directory. For most systems this will be '/u', but this may vary from system to system. The Root filter displays file starting with the file system root (i.e. '/'). If you want additional filters, you could right-click 'USS Files' and from the context menu, select Add Filter. Note that like the other filters, you can specify multiple criteria by right-clicking the filter, and from the context menu, selecting Change or Properties...In future lab exercises, we will be working with your home directory. Since there is a built-in filter to display the contents of your home directory, we will not need to add any USS File Filters for the upcoming lab exercises.

## 1.6 Optional - Explore TSO Commands

This is an optional exercise.

If you are interested, you can explore the TSO Commands feature of RDz Users can launch TSO session from Remote Systems view, they could have multiple TSO Sessions up at the same time. The New TSO Commands UI is based on USS Shells by

RSE. Let's see one example below.

Using the z/OS projects view and Remote System perspective, be sure that you are connect to the z/OS. If not connected, right-click on demomvs and select connect.

64. Scroll down to see the TSO Commands note. Right Click on it and select → Launch TSO as seen below:



#### 65. The Remote shell view will open

😺 Remote Error List (🕀 z/OS File System M 🛱 Property Group Ma 🕼 Remote System Det ( क्षि Servers	🖪 Remote Shell 🕺 🦵 🛙
	- 🖉 🔳 🗶 🛯
TSO-demomvs	
Specify a TSO command to run	
	Þ
Command	<b>\</b> \

66. You will be able to execute commands, like the command time seen below:



67. Also in the command line you will be able to use the Ctrl + Space to have the content assist, as seen below. Type I and press Ctrl + Space and select LISTALC:

👩 Remote Error List 🖶 z/OS File System M 📴 Property Group M	a 🕼 Remote System Det 🙌 Servers 🖫 Remote Shell 🕱 🦳 🗖
	Z 🗏 🧏 😼 🛛
TSO-demomvs	
Specify a TSO command to run	
E >time LINK	DISPLAY ACTIVE DATA SETS.
IKJ5665 IIIISTALC	CEMBER 18,2(
LISTCAT	
LISTDS	
Command Li	

#### 68. The result will the one below



69. You also can use the icons below and perform some activities..



- 70. Right click on your remote project or anywhere within your demomvs remote connection. Select Host Connection Emulator Support from the context menu. You will be connected to z/OS.
- 71. Resize the window so you will be able to better see the 3270 black screen. Just Doubleclick on the blue title demomvs.hce
- 72. Depending on the demosystem you use your screen will look similar to

demomvs.	hce 🗙 🔪	~					
Current host c	Current host connection profile is: /HostConnectProjectFiles/demomvs.hce						
							****
	*		Boebli	ngen TMCC			*
	*		z/OS	1.7.0			*
	* Enter	an applic	ation nam	e or a VTA	M LOGON C	ommand.	*
	* Exampl	e: TSO u	serid or	LOGON AI	PLID(appl	id)	*
	*******	********	*******	*********		********	****
	TSO	- TS	SO/E (TSOM	n for LOGO	ON with Mo	del n=2,3,	4,5)
	CIC	s – ci	CS TS 3.1				
	IMS	- IN	IS V9				
	z/0	S SYS1	z/OS Summ	er School	System		
	IPADDR:	192.168.1	7.8				
	PORT :		02221				
	===>						
DE 1	DE2	DE3	DE4	DES	DE6	Enter	DA1
<u></u>	112					Linter	TAL
PF7	PF8	PF9	PF10	PF11	PF12	Clear	PA2

73. Note that all PF keys are represented by buttons. Of course you can also use your keyboard. The default keys in addition to the PF keys are:

[Clear] – Esc

[Enter] – Return or Ctrl key

You can change those keys via Window  $\rightarrow$  Preferences (from the menu bar) and then expanding General and selecting Keys.

74. For example to assign the Pause key to the [Clear] command as used in IBM Personal Communications select the [clear]-command, put the cursor to the Binding field, then hit Pause and at last hit Apply (which is not shown in the picture).

L. L			
e filter text	Keys		
General	Scheme: Default	•	
Editors	Command A	Rinding	When
Error Reporting	Command A	Binding	when
Keys	[bidilayer]	Ctrl+N	In Host Connec
Network Connections	[clear]	Pause	In Windows
Perspectives     Search     Search     Security     Startup and Shutdown     Web Browser     Workspace     Workspace     Agent Controller     Ant     Auto Comment     Backward Compatibility     Bidirectional Development     BMS Map Editor     COBOL     COBOL     Connections     Crystal Reports     Data Management     Ecore Diagram     Enterprise Service Tools     File Manager     Help     Importer     Install/Update     Java	[enter]         [eraseeof]         [erasefid]         [erinp]         [newline]         [pa1]         [pa2]         [na3]             Qopy Command       Unbind Command             Name:       [clear]         Description:       Hit Pause key         Binding:       Pause         When:       In Windows	Restore Command	Conflic Comr

- 75. Click OK to save your changes and exit.
- 76. You can now optionally follow the screen instructions to logon to TSO (usually done by typing "TSO" and hitting [Enter]). Use your z/OS assigned user id USER##. Do not forget to logoff properly before closing the emulator.

# 2 Work with remote files

Working in eclipse is traditionally realized with projects. For z/OS this means that you will link your assets into a project structure: into z/OS projects. These projects can have USS or MVS subprojects, depending on whether you want to work with files residing in USS or members in MVS. In both cases your assets stay on the host and are only logically linked into sub projects.

By associating Property Groups with those sub projects you can bring your members or files into a context with compile, link and execution properties.

This chapter will guide you through the process of creating a remote project and setting its properties for different languages and runtimes.

## 2.1 Retrieve and import property group

Property Groups are a new feature of RDz 7.5. They contain the definitions for the z/OS Projects to specify build properties.

In earlier versions, properties needed to be defined for each project (respectively subproject) separately. Now all properties are kept separately and managed in so-called Property Groups which are associated with a remote connection. They can easily be imported, exported, copied, modified and associated with projects or members.

We will import the Property Groups for our lab which are stored in a XML file on the host. We will drag and drop this file from our remote system (USS) into a local project and import it.

- 77. Open the Navigator View by selecting Window → Show View → Other... Expand General and choose Navigator. Move the Navigator View if you like it in another position.
- 78. Create a new general project by selecting File → New → Project and then General → Project. Name it "localFiles". This project is just a place to store our local files.
- 79. Go to the Remote Systems View and expand USS files and Root. Navigate to /local/zoaws/props (this is a directory on your demomvs, not local on your PC!)
- 80. Select remoteProps75.xml and drag and drop it into your localFiles project in the Navigator View (Note: this will only work with the Navigator View).
- 81. To ease your search for this file in the file system later, we will copy the location. Right click your properties file and select Properties. Mark the path specified for Location, right click and select Copy to copy the path.

Info		(= + )	÷ -
<u>P</u> ath: <u>T</u> ype: Location:	/localFiles/batchCobol.xml File (XML) C:\Documents and Settings\A	ıdministrator∖My	
	Documents \workspaces \work	space_rdzV71_zHero\localFiles\batchCobr	ol.xml
<u>S</u> ize:	9964 bytes		
Last <u>m</u> odified: November 21, 2007 11: <u>R</u> ead only		Cut	
		Сору	
		D 1	

Now our properties file is available and ready to import.

82. To import the Property Groups, locate the Property Group Manager View at the lower panel group, rightclick on demomvs and select Import from the context menu.

🐼 Remote Error List   z/OS File Syste	em Mapping 🕞 Property Group Manager 🛛 📕 Remote
Name	Description
LOCAL Import	

83. Now either paste the content of your clipboard to the file destination field or use the browse option to locate the file.
84. Check ÁLL Property Groups and click OK.



You now see the property group as child of the remote system connection.

# 2.2 Create an remote project

- 85. Select File  $\rightarrow$  New  $\rightarrow$  Project... from the menu bar
- 86. Scroll down, Expand z/OS, select z/OS Project and click Next

🖸 New Project	
Select a wizard Create a new z/OS project	
<u>W</u> izards:	
type filter text	
Web  Workstation COBOL or PL/I  C/OS  WVS Subproject  JOS Project  C/OS Project  Examples	
Show All Wizards.	
O < Back [Next > Enish	Cancel

87. On the New z/OS Project panel, name the project "hostProject" and select the "Do not create a subproject now" radio button.

🔘 New z/05 Project	
z/OS Project Name	
Create a z/OS Project	
Project name: hostProject	
Subproject	
Do you also want to create a subproject now?	
Create an MVS subproject	
Create a USS subproject	
Do not create a subproject now	
< Back         Next >         Einish	Cancel

88. Click Finish. This will create a z/OS parent project.

We will now create an MVS sub project.

- 89. You can define the MVS project only when you are connected to the system. Make sure you are connected in your Remote Systems view.
- 90. Go to the z/OS Projects View, right-click hostProject that you just created and select New → MVS Subproject from the context menu.
- 91. Depending on which lab you are going to work on check the Property Group with the name in the following table:

	Batch	CICS	IMS
COBOL	batchCobol75	CICSCobol75	
PLI			
C/ C++			
Assembler			

#### Table 1 - available Property Groups for different languages and runtimes

92. Name your MVS sub project like the name of the Property Group (the screenshot shows an example) and select your userid as high level qualifier. Click Finish.

New MVS Subproject		
MVS Subproject Name and Create an MVS Subproject	l Location	MUS
Host Short Name: zHero Project Name: hostProject Subproject Name: batchCobo Subproject Type: MVS High-Level Qualifier: USER##	175	4
Select a property group to associ Name ZHeroBatchCobolProps	iate with the new subproject.  Description batch Cobol on zHero System CICS 3.2 Cobol on zHero System batch Cobol on zHero System	New Edit
0		Cancel

93. You should see an MVS Project in your z/OS Projects view. If you click on the + sign to expand the project, you'll see that is empty.

We will review the imported properties later.

This lab will continue with batch COBOL programming to demonstrate more RDz features and functions. Please look at Table 1 - available Property Groups for different languages and runtimes - to see which other languages and runtimes are currently available for this workshop. Please contact your lab administrator to receive other lab material.

# 3 COBOL

In this chapter you will learn how to work with remote Cobol in batch. You already defined an MVS project and imported the property group – now we will have a more detailed look at the Property Group itself.

You will start with a very simple HelloWorld program (mostly generated) that you will edit, compile, link and run. Later on you can have a look at some more complex programs for debugging and other features like dependency check.

# 3.1 Verify project properties

Before you start to work with remote COBOL programs let's have a look at the property group you already associated with your remote MVS subproject.

- 94. In the Property Manager Group, expand the remote System demomvs, then Right-click your batchCobol75 property group and select Edit.
- 95. At first a short overview about the title and a description is given, click next to go to the next parameters

(	🕽 Edit Prope	rty Group	
E	dit Proper	ty Group Information	
	Edit information	on for property group batchCobol75	
	Property are		
	Name:	hatchCohol75	
	r von nor		
	Description:		
	System:	demomvs	
	?	< Back Next > Finish	Cancel
_			

96. The next panel offers you to select the property you want to set. This is one of the major improvements of the new version regarding the project property groups. Please verify that the following options are ticked, then hit Next:

💽 Edit Property Group	
Select Properties for Property Group	
Select the properties to be included in the property group.	
Property group batchCobol75	
Assembler Settings	
COBOL Settings	
I_ C/C++ Settings	
PLI Settings	
Screen Maps	
BMS Settings	
MFS Settings	
Other	
ICL Job Card and Data Set	
✓ Link Options	
Run-time Options	
< Back         Next >         Einish	Cancel

# 97. The COBOL Settings panel should be next, expand ELAXFCOC procedure, select COBOL and click Edit Step....

Edit Property Group					_ 🗆 ×
Edit Properties in Pr	operty Group				
Edit the properties in the	property group				
- COBOL Settings	COBOL Settings				
- Link Options	🗌 Use DB2 (contai	ns EXEC SQL s	tatements)		
Run-time Options	Use CICS (conta	ains EXEC CICS	statements)	CICS	
	Procedure Name	Step Name	Status		Edit step
	± ELAXFCOP		Disabled Disabled		Add step
	ELAXFCOC	COBOL	Enabled		Remove step
		CODOL			
					Down
					Enable procedure
					Disable procedure
	•				
0					Finish Cancel

Note that we could enable and disable the procedures that must be executed before the COBOL compilation done by ELAXFCOC if necessary. The default supplied procedures (but disabled) are:

ELAXFCOP - invoke separate DB2 pre-compiler

ELAXFCOT - invoke separate CICS translator

But for this lab: do NOT invoke them.

By setting the checkmarks next to "Use DB2" or "Use CICS" you can invoke the integrated translator. Also, do not check them because we are working in Batch.

98. The COBOL Compile Step Options dialog will open. Review the settings. <HLQ> will be replaced by the userid you specified during creation of your MVS project. Select OK to confirm your changes.

Cobol Compile Step Options	×	
Compile Procedure Name:		
ELAXFCOC		
Compile Procedure Step Name:		
COBOL		
Compiler Options:	Compiler option TEST	
	which will create a so	called side file in
Listing Output Data Set:	the USER##.COBOL.	SYSDEBUG
	data set. This file will	be required later
Debug Data Set:	at runtime by the IBM	Debug 1001 to
<hr/>		
Object Deck Data Set:		
<hlq>.COBOL.OBJ</hlq>	Data set to place obje	ect
Copy Libraries:		
<hlq>.COBOL.COPYLIB</hlq>	Data set to search for	copybooks
Support Error Feedback		
Data Set Qualifier for Compiler Errors:		
<hlq>.ĖRRCOB</hlq>		
Additional JCL:		
//******* ADDITIONAL JCL FOR COMPILE HERE ******		
	<b>v</b>	
•	Þ	
2	OK Cancel	
	Cancer	

99. Select JCL Job Card and Data Set. <JOBNAME> will automatically be replaced by your userid during submit.

O Edit Property Group		
Edit Properties in Pro Edit the properties in the	property Group	
····· COBOL Settings ····· JCL Job Card and [	JCL Job Card and Data Set	
- Link Options Run-time Options	JCL Job Card: // <jobname> JOB , // MSGCLASS=H,MSGLEVEL=(1,1),TIME=(,4),REGION=70M,COND=(16,1)</jobname>	LT) V
<u> </u>	JCL Data Set: <pre></pre>	
0	< <u>B</u> ack (Next > Einish	Cancel

100. Select the Link Options. Expand the + besides ELAXFLNK, click on LINK and click on Edit Step...

O Edit Property Group				<u>_                                    </u>
Edit Properties in Prope	r <b>ty Group</b> erty group			
COBOL Settings JCL Job Card and Data S Link Options Run-time Options	Link Options Procedures and Steps Procedure Name ELAXFLNK	JCL Substitution	Step Name	Edit step Add step Remove step
0	<	Back Next >	Einish	Cancel

101. Adjust the link options to match the following picture and select OK to confirm your changes.

Link Step Options	×
Linkage Editor Procedure Name:	
ELAXFLNK	
Linkage Editor Procedure Step Name:	
LINK	
Link Options:	
Link Libraries:	All other required link libaries are set in
<pre><hlq>.COBOL.OBJ</hlq></pre>	the ELAXFLNK proc
Append to: Position	
☑ Use specified link instructions:	
INCLUDE SYSLIB(EQADBCXT)	
	-
र	<del>ا</del>
Load Module Location:	
<hlq>.COBOL.LOAD</hlq>	Data set to place load module
Additional JCL:	
//****** ADDITIONAL JCL FOR LINK H	IERE *****
<u>.</u>	F
0	OK Cancel

102. Go to Run-time Options, expand ELAXFGO, click on RUN and click "Edit step..."

💽 Edit Property Group			
Edit Properties in Proper	rty Group erty group		
COBOL Settings JCL Job Card and Data S Link Options Run-time Options	Run-time Options Procedures and Steps JCL	Substitution	
	Procedure Name	Step Name RUN	Edit step Add step Remove step
0	< <u>B</u> ack	Next > Einis	h Cancel

103. Make sure that "Run in Batch" – not "Run in Batch with Debugger" is selected. We will debug later

Run-time Step Options
Run in batch
C Run in batch with debugger
Run Procedure Name:
ELAXFGO
Run Procedure Step Name:
RUN
Program Parameters:
Run-time Options:
J
Select the order of Run-time Options and Program Paramteters
Program Parameters / Run-time Options
C Run-time Options / Program Parameters
Additional JCL:
//****** ADDITIONAL RUNTIME JCL HERE ******
OK     Cancel

You have now seen all properties of the group that we are going to use in the next steps. To apply a property group we will associate it with a member, data set or project.

# 3.2 Create a COBOL HelloWorld program using zAPG

- 104. Select your COBOL dataset USER##.COBOL out of the z/OS Projects view.
- 105. Select File  $\rightarrow$  New  $\rightarrow$  Other.. and choose COBOL  $\rightarrow$  Cobol Program. Click Next.
- 106. Name your program and member HELLOW and place it in **USER##**.COBOL PDS.

💽 New COBOL Program			
COBOL Program Create a new COBOL prog	ram		COBPL
Program Name: HELLOW Author: USER## Target • Partitioned Data Set			
C Sequential Data Set			
Partitioned Data Set	remoteCobol	<b>_</b>	
Host Code Page	IBM-037		
Data Set name:	USER##.COBOL		Browse
Member:	HELLOW		
Add comments to gene	rated program		
Open <u>S</u> nippets view wł	en finished		
?	< <u>B</u> ack <u>N</u> ext 3	> <u>F</u> inish	Cancel

Click Next.

107. On the next panel you can specify features that will add code to your new program. Do not select any, but click on the "Define Features" button to have a look how a feature could look like.

💽 New COBOL Program	
COBOL Program Features	
Specify features that will be used in the COBOL program.	СОВРЬ
Which features would you like to add to the program?	
Use BMS Maps Invoke CICS commands Use SQL statements Handle SQL error return codes	
	Define features
⑦ < <u>Back</u> <u>N</u> ext >	<u>F</u> inish Cancel

108. If you click on a feature the tabs below will show you what code will be added at which division or section. You could also create new or modify features here. Click Cancel to dismiss the dialog.

Preferences (Filtered)		
type filter text	Features	$\leftarrow \star \Rightarrow \star$
E COBOL	Choose and customize features available to COBOL programs. Features: (check to enable)	
	✓       Use BMS Maps         ✓       Invoke CICS commands         ✓       Use SQL statements         ✓       Handle SQL error return codes         Environment       File         Working Storage       Local Storage         Linkage       Statements to generate in this section:	New Delete Rename
	+-*A-1-B+2+3+ EXEC SQL INCLUDE SQLCA END-	-4+- EXEC.
	Restore <u>D</u> efaults	<u>A</u> pply
0	ОК	Cancel

109. Click Finish and your program will be created. The editor opens automatically. A basic skeleton was created for you.

110. The preferences for this skeleton have been defined for your workspace. Click Window → Preferences and navigate to COBOL → Code Templates and you will see what led to creation of your program. You could use these templates to spread coding guidelines for example.



#### 3.3 Add resources to your project

To make the new member available to your remote project, you will need to add it.

- 111. Locate your newly created member in your Remote Systems View and richt click it. (If you don't see your member, right click the filter or the corresponding data set and select Refresh).
- 112. Select "Add to Subproject..." from the context menu.
- 113. Accept parent project name hostProject and sub project name batchCobol75 and click Finish to add the member to the project.

💽 Add		
Add Resources Add selected resou	rces to subproject.	
Project Name: Subproject Name:	hostProject batchCobol75	<b>-</b>
	,	
0	Einish	Cancel

114. Switch to the z/OS Projects view and you will see that USER##.COBOL has been added to the batchCOBOL project. The z/OS Projects view should look like the following:



115. Before we can proceed, we might need to apply a Property Group, which is new to version 7.5. Therefore right click the project batchCobol75 -> Property Group -> Associate Property Group. Tick the property Group and then hit OK.

O Associate Property Group		
Name	Description	
0	OK Can	cel

# 3.4 Exploring the Editor

Code assist helps with the completion of code and is activated by pressing Ctrl + Space when in the content area of the z/OS LPEX editor.

- 116. In your HELLOW program navigate to the empty line below PROCEDURE DIVISION. Place your cursor in column 12 (B) (from beginning of the line hit TAB 2 times).
- 117. Due to the fact, that in Cobol the column in which the line of code starts is depending on strict rules, you might enable a margin line. You might enable this in the preferences  $\rightarrow$ LPEX Editor  $\rightarrow$  System z LPEX Editor  $\rightarrow$  COBOL Parser

Tick the options like displayed and hit Apply and OK.

Preferences		<u>- 🗆 ×</u>
type filter text	COBOL Parser	$\leftarrow \star \Rightarrow \star \star$
… Autosave     … C/C++ Parser     … COBOL LPEX Editor     … COBOL Parser     … Controls     … Find Text     ⊕ HLAsm Parser	<ul> <li>Enable Margin A line at column 8</li> <li>Enable Margin B line at column 12</li> <li>Enable Margin R line at column 73</li> </ul>	Ŗ

The result are the following margins:

HELLOW.d	ых	- 8
Line	11 Column 14	4 Insert
	+-*A-1-B- <mark>-</mark> +	2+3+4+5+6-
000001	*******	************
000002	*	
000003	*	
000004	********	*******
000005		
000006		
000007		
800000	IDENTIFIC	ATION DIVISION.
000009	PROGRAM-I	D. HELLOW.
000010	AUTHOR. zi	user11.
000011		
000012		
000013	ENVIRONME	NT DIVISION.
000014		
000015		
000016	DATA DIVI	SION.
000017		
000018		
000019	PROCEDURE	DIVISION .

118. Type "DI" and press Ctrl+Space. A window will show all suitable Cobol commands starting with "DI".

5	*HELLOW	сы 🗙		- 8
	Line 2	20	Column 14 Insert 2 changes	
		*	*A-1-B-++2+3+4+-	5+
	000019		PROCEDURE DIVISION .	<b></b>
	000020		DI	
	000021		REC DISPLAY	
	000022			
	000023		RE DIVIDE - NOT ON SIZE ERROR - END-DIVIDE	
	000024		BUVIDE - ON SIZE ERROR - END-DIVIDE	
	000025		BUDIVIDE - ON SIZE ERROR - NOT ON SIZE ERROR - END-DI	_
	<u> </u>		-	
	<u> </u>			

- 119. Press Enter to select the first one (DISPLAY) and insert the string "Hello World".
- 120. Use Ctrl+ Enter to create a new line (like ISPF newLine command).
- 121. In the next line we want to write "STOP RUN.", use content assist as well.

5	*HELLOW.	ых
	Line 2	1 Column 18 Insert 6 ch
		+-*A-1-B+ <b>-</b> -2+3
	000019	PROCEDURE DIVISION .
	000020	DISPLAY "Hello World"
	000021	STOP r
	000022	ABC RUN

122. Explore the different code completion possibilities. Press the Esc key to dismiss the popup list. For COBOL verbs that require additional parameters (identifier in COBOL terminology), code assist will present data items defined in the working storage section as selections in the pop up list.

Finally your PROCEDURE DIVISION should look like

```
PROCEDURE DIVISION .
DISPLAY "Hello World"
STOP RUN.
```

# 3.5 COBOL Syntax Checking

Before submitting a job to the z/OS system to compile the COBOL source file, you can perform a syntax check to ensure a clean compile.

You will deliberately introduce an error to illustrate the error feedback facility.

Since the new version 7.5 a new *real time* syntax check is implemented. When you do a typo a small exclamation mark is shown at the first column in front. This is inherited from the the distributed application development within eclipse.

- 123. Overtype DISPLAY with DSPLAY (NOT DISPLAY) to force an error and save your changes (CTRL + S).
- 124. You now see already the yellow small exclamation mark in front of the line. At the right margin, please be aware of the small yellow-white flag. The margin represents the whole document, and you have a overview of the errors and hints of the document.

) HELLOW.d		3
Line 2	21 Column 51 Insert	
	+-*A-1-B+2+3+4+5	
000015		
000016	DATA DIVISION.	
000017		
000018		
000019	PROCEDURE DIVISION .	
000020	DSPLAY "Hello World"	
000021	STOP run.	
4		
	<u> </u>	

125. Now hover over the exclamation mark at the left – a hint is displayed. You can as well hover over the small flag at the right side of the editor.

	HELLOW.cb		
	Line 2	20 Column 13 Insert	
	-	+-*A-1-B-++2++3++4+5++	6
	000015		•
	000016	DATA DIVISION.	
	000017		
	000018		
	000019	PROCEDURE DIVISION	
4	"DISPLAY"	expected instead of this input Hello World"	
6	000021	STOP run.	Ţ
	•		▶□□

126. If you do not like to have the real time syntax check activated, you might change the settings for the Cobol Parser Settings in Window → Preferences → and in the navigation tree LPEX Editor | System z LPEX Editor | Cobol LPEX Editor

Preferences		
COBOL	COBOL LPEX Editor	$\leftarrow \bullet \bullet \bullet \bullet \bullet$
COBOL COBOL XML Converters COBOL XML Converters COBOL Requester Te COBOL Requester Te COBOL COBOL COBOL COBOL COBOL LPEX Editor COBOL LPEX Editor COBOL LPEX Editor COBOL Parser C:\Workspaces\RDz 75	"Enable editor annotations and tooling" in order syntax checking of COBOL files via warning an "Enable tooling only" to disable editor annotation actions in the context-menu enabled (e.g. "Op "Refactor"). "Disable editor annotations and to editor annotations and context-menu actions. COBOL Parser Setting © Enable editor annotations and tooling © Enable tooling only © Disable editor annotations and tooling	r to receive real-time notations in the editor. ons but leave the editor en Declaration", poling" to disable both the

The next steps describe the explicit local syntax check.

- 127. Right click anywhere within the code or right click on HELLOW.cbl in your z/OS Projects View and select Syntax Check → Local.
- 128. In the dialog box specify that you want to do a syntax check only.

O Check Dependencies?
You might want to check the dependencies before proceeding with the syntax check.
<ul> <li>Syntax check only</li> <li>Refresh dependencies and then perform a syntax check</li> </ul>
OK Cancel

Note: For a local syntax check local resources will be used. This means you can save CPU by performing local tasks.

129. The TPF Toolkit Console will provide Feedback about the compile process. It should return a reason code of 12.

🐼 Remote Error List 🕪 z/OS File System Mapping 🛱 Property Group Manager 💷 Remote Console 🛛 📕
ProgramFiles=C:\Program Files TMR=C:\DOCLIME::1\ADMINI.:1\COCALS::1\Temp
SystemDrive=C:
CLASSPATH=C:\Program Files\IBM\WebSphere MQ\Java\ib\com.ibm.mqjms.jar;C:\Program Files\IBM\WebSphere MQ\
COMMONProgramFiles=C: (Program Files (Common Files COBPATH=C: \Program Files \IBM\SDP70Shared \plugins \com.ibm.ftt.ui.views.navigator 9.0.0.v200810101454
LANG=en_US
NLSPATH=C:\Program Files\IBM\SDP75\messages\%L\%N LOCPATH=C:\Program Files\IBM\SDP75\locale
SYSLIB=
Output:
staErr: Return Code: 12

130. Click on Remote Error List view (in the bottom) to check compilation errors.

Í	0	Remot 🕅	🛨 z/OS Fi 🖳 Proper	🖳 Remot 🗋 📕	Remot	해 Server	🖳 Remo	t 🗋 🔚 Snip	pe] 🗆 🗖
	Filter matched 1 of 1 messages 🛛 💥 🧍 🐕 🧏 🗐 🏹								
		ID	Message	Seve	Line	Location	n		Host Name
	8	IGYPS2072	IGYPS2072-S "DSPLAY" wa	as invalid 2	20	hostPro	ject/remot	eCobol/	demomvs

131. Double click on the error message. This should bring you to the editor. Note that a red mark was placed next to the error. If you use your mouse to hover over this mark the error message is displayed again.

	HELLOW.d	
	Line 2	21 Column 13 Insert
		+-*A-1-B-++2+3++4+5++6
	000015	
	000016	DATA DIVISION.
	000017	
	000018	
	000019	PROCEDURE DIVISION .
(All h	IGYPS207	2-S "DSPLAY" was invalid. Skipped to the next 1d."
Ľ	expected	lingtead of this input
	•	

132. Fix that returning the old version that had the correct DISPLAY statement.. Click CTRL + F4 or CRTL + w to close the editor.. 133. Perform another Syntax Check to ensure that the compilation is now clean. The error message at the problems page must be gone.

_	<u> </u>			1 0	<u> </u>									
Ø	Remot	×	<table-cell-rows> z/OS F</table-cell-rows>	i 🗟 Proper	📃 🔜 Remot	🗋 📕 Rem	iot ) 해당	Server	R: R	emot	) 🗈	Snipp	pe	
Filte	r matche	ed 0 of (	0 messages						×	Þ,	≱≝	-≱i ->1	<b>R</b> ,	[] ▽
	ID		Message			Seve	Line	Location	n				Host	Name
L														
L														
L														
L														

134. Close all opened editors if still opened. (CTRL + F4)

#### 3.6 Compile/link/execute the remote COBOL program

In this part you will learn how to compile your programs, link them and execute them on z/OS.

#### 3.6.1 Using Project Build

The Project Build Feature will build all Members contained in your project automatically. At the moment, your whole Cobol dataset is contained in the project, so for this section we will remove it and add the Cobol member only.

135. Go back to the "z/OS Projects" View, right click your remoteCOBOL subproject and select "Rebuild Subproject" from the context menu. The result will look similar to



136. To run your project, right click your load module USER##.COBOL.LOAD(HELLOW).exe and select "Run" application from the context menu. A popup window will inform you that a job has been submitted. This job runs your load module.

🖸 Job submission confirmation 🛛 🛛 💌							
į)	WZID0805I: 'JOB04062' has been submitted to host 'demomvs'. Use JES on the Remote Systems view to view its output.						
	OK	]					

137. To review the output go to the "Remote Systems" View and expand JES and an appropriate filter (at least the "My Jobs" filter should display your results) to review your job output.



### 3.6.2 Using JCL Generation

You will use a feature called JCL generation, where the properties of your Remote z/OS project are used to generate JCL for Compile only, Compile and Link or Compile, Link and Go.

We are using this feature for demonstration purposes. In real world of course you will either use you company's provided compilation procedures or take advantage of a source code management system like SCLM or Endeavor.

Now that you have a successful syntax check of your COBOL program, you can generate the JCL (Job Control Language) that will be used to create the executable on your z/OS system.

- 138. Using the z/OS projects View, Right-click on HELLOW.cbl and select Generate JCL → For Compile Link Go.
- 139. On the JCL Data Set and Member Name window, notice that the JCL Data Set Name is set to the value you specified for your project settings.

O JCL Data Set and Member Name							
Job Name:	USER##						
JCL Data Set Name:	USER##.COBOL.JCL						
Member Name:	HELLOW						
0	OK Cancel						

140. Click OK. You should see the message below.

💽 JCL 🔘	Generation Message	x
į	WZID0913I: Generated JCL is in the PDS 'ZUSER11.COBOL.JCL'.	
	OK	

141. Go to your z/OS Projects view and you will see that HELLOW.jcl was generated.

8/10/2009

142. Submit your JCL (Right click → Submit), go to your Remote Systems View and expand the Job under JES.



143. Double click on GO:SYSOUT to see the program output.



#### 3.6.3 Using Menu Manager

You can use a feature called Menu Manager to parameterize JCL with variables from your workbench. Please refer to the separate Menu Manager Lab.

# 3.7 Debugging Cobol

We will now have a look at a little more complex program and execute it in debug mode. The member REGI0A.cbl has been copied to your Cobol dataset.

To execute in debug mode, we have 3 options:

- Modified property group for the JCL generator
- Right click menu of a load module within a MVS project
- Exits
- 144. Either create a new MVS subproject or empty your existing project (careful, don't delete, select "remove from subproject" from the right click context menu) and add REGI0A.

#### 3.7.1 Some more editor goodies

As REGI0A is a little more complex we can explore some more editor functions.

145. Right click on the variable "FIELD-A" and select "Open Declaration". The focus will jump to the declaration.

5	REGIOA.cbl 🛛	3 📄 HELLO	OW.cbl				
	Line 37	Colu	ımn 38	Insert			
	-+	-*A-1-B	+2	+3-	+	-45	;
	000030						<b></b>
	000031	LINKAGE	SECTION				
	000032	PROCEDU	RE DIVIS:	ION.			
	000033	010-ІМІ	FIALIZAT:	ION.			
	000034	*	Initiali	ze Progra	m-work-	Save	
	000035		DISPLAY '	"Program	REGIOA .	2010	
	000036	1	MOVE 2 TO	D BRANCHE	LAG.	Cut	
	000037	1	MOVE 'AAA	AAAA' to	FIELD-A	Сору	
	000038	1	MOVE 'BBI	BBBB' to	FIELD-E	Paste	
	000039	1	MOVE CCC	CCCC' to	FIELD-C		
	000040	1	MOVE "Ent	terprise	Transfc	Select	
	000041	1	MOVE "WSI	ED - OUTE	UT" to	Selected	
	000042	1	MOVE "RE(	GIOB" to	program	Deselect	
	000043	020-LOG	IC.			Filter view	
	000044		CALL pro	ogram-to-	call US	Show all	
	000045	1	move 66 1	to value1	•	Driow di	
	000046	1	move 1 to	o receive	d-from-	Source	
	000047		divide va	alue1 BY	receiv	Refactor	
	000048		DISPLAY	"The resu	lt is .	Open Declaratio	in

146. Mark the line saying PERFORM 030-SEEYA right click it and select "Open Perform Hierarchy"

Line 52	Column 37 Insert	Save
	+-*A-1-B+2+3+	2010
000049	IF BRANCHFLAG > 1	Cut
000050	CALL 'REGIOC' USI	Сору
000051	DISPLAY "BRANCHFL	Paste
000052	PERFORM 030-SEEYA	Colorat
000053	ELSE	Select
000054	DISPLAY "BRANCHFL	Selected
000055	PERFORM 040-GOODB	Deselect
000056	GOBACK.	Filter view
000057	030-SEEYA.	Show all
000058		51097 01
000059	DISPLAY "PROGRAM IS EN	Source
000060	040-GOODBYE.	Refactor
		Open Declaration
•		Open Perform Hierarchy

147. The Perform Hierarchy Window will open and the focus will jump to the declaration.



148. Click on the buttons to toggle between Performer and Performee Hierarchy

#### **3.7.2 Modified property group for the JCL generator**

Your generated JCL can be modified to initialize debugging. Instead of modifying the parameters, we will create a second Property Group for Debugging.

- 149. Navigate to the Property Group Manager and right-click on the existing Property Group batchCobol75 and select Copy...
- 150. Now right-click the new Property Group "Copy of batchCobol" and open the edit panels. Before proceeding rename the Property Group to a more meaningful name like batchCobol75 Debugging. Hit Next.

All Properties are now identical to the template. We now only modify the parameters for debugging.

151. On the Run-time Options panel, expand the + next to ELAXFGO, select RUN and click "Edit Step".

💽 Edit Property Group		
Edit Properties in Property	Group	
Edit the properties in the property	group	
COBOL Settings JCL Job Card and Data Set Link Options Run-time Options	Run-time Options       Procedures and Steps       JCL Substitution	
	Procedure Name Step Name	Edit step
	RUN	Add step
		Remove step
		Up
		Down
		•
0	< Back Next > Einish	Cancel

152. On the Step Options Panel, select "Run in batch with debugger" and make sure that "Program Parameters / Run-time Options" is selected as order. Click OK. And then Finish to save changes.

O Run-time Step Options	4
C Run in batch	
Run in batch with debugger	
Run Procedure Name:	
ELAXFGO	
Run Procedure Step Name:	
RUN	
Program Parameters:	
Run-time Options:	
J	
Select the order of Run-time Options and Program Paramteters	
Program Parameters / Run-time Options	
C Run-time Options / Program Parameters	
Additional JCL:	
//****** ADDITIONAL RUNTIME JCL HERE ******	
OK Cancel	

153. Now do a right click on REGI0A.cbl and select Property Group → Associate Property. Select the Property Group you just prepared for Debugging:

Associate Property Group		
Property group		
Name	Description	
batchCobol75		
im ✓ batchCobol75-Debugging		
-		
0	ОК	Cancel

154. To generate the JCL right click REGI0A.cbl and select Generate JCL  $\rightarrow$  For Compile, Link and Go.

155. Open the generated JCL and scroll to the end of the JCL to look at the runtime options. RDz inserted a TEST runtime parameter specifying your IP address. This was done because you asked for remote debugging in the Run-time Step Options.



156. Submit the Job. After a moment, you are asked, if you want to switch to the Debug Perspective

You can now either cancel debugging to try the other options or you can directly jump to chapter "3.7 Debugging Cobol" on page 64 and try the other options later.

#### 3.7.3 Right Click Menu

- 157. Initiate a project build on the project that you added REGI0A to. Your load member should be available now (USER##.COBOL.LOAD(REGI0A))
- 158. From your project, right click the load member and select "Debug Application" from the context menu.

You can now either cancel debugging to try the other options or you can directly jump to the debugging instructions and try the other options later.

#### 3.7.4 User Exits

This feature is currently not available.

#### 3.7.5 Using Menu Manager

You can use a feature called Menu Manager to parameterize JCL with variables from your workbench. Please refer to the separate Menu Manager Lab.

# 3.7.6 Debug

159. The window below will ask if you want to switch to the Debug Perspective. Click Yes.

💽 Confi	irm Perspective Switch				
This kind of launch is configured to open the Debug perspective when it suspends.					
	This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.				
	Do you want to open this perspective now?				
Remember my decision Yes No					

160. Now the Debugger Perspective is opened and you will see the image below.

Debug - RemoteSystemsTempFiles/DebugView	vFiles/8001_11160/ZUSER11.COBOL.SYSDEBU 💶 🗖 🗙
File Edit Navigate Search Project Run Window	Help
📬 • 🗒 🖻   💆   55 • ☆ • O • 🚱 • O • .	<ul> <li>A Debug</li> <li>Java</li> <li>Data</li> <li>Z/OS Projects</li> </ul>
] <b>29                                   </b>	
🏇 Debug 🔀 👯 Servers 🗖 🗖	(≫= Vari 🛛 💊 Bre 🔐 Reg 📴 Mon 🕯 Mod 🖓 🗖
<ul> <li>REGIOA [Remote Compiled Application]</li> <li>Redium Cos/390(R) Connection: 192.168</li> </ul>	Name   no local variables are available for the selected stackfra
Image: Stress	Outline X     Outline X

161. Close the outline view that is not be useful in this debugger.

162. Click twice on the icon Step Over 📀 (or F6) located in the Debug view until you stop in the DISPLAY command:

Note: If the icons are disabled, click on REGI01:01 under Thread:1



163. Double click to the left of line number 000040 to add a breakpoint. The symbol (2) appears. Note that the breakpoint will also be added to the Breakpoints View (upper right).

*	F Debug 🛛	해 Servers		(x)= Vari 🔍 Bre 🔀	lili Reg 💇 Mon	ඩ් Mod 🗖 🗖
				📉 💥 💥 🖂	🔌 🧏 📴 🔅	ŧ 🛛 😫 🎽
X	🤅 🕩 11 🔳	8 3.00	. r =		SER 11. COBOL. SYSD	EBUG(REGI0A) 4
7	ं । 😪 🔊 -					
E	REGIOA [R	emote Compiled Applicat	ion]			
	Platfor	m: OS/390(R) Connect	ion: 192.168			
	🖻 🛷 Th	read:1 (Runnable )				
		REGIOA:01				
	Proces	s: 500232616 Program:	REGIOA			
•			D.	•		► I
	) REGIOA.jd	SUSER 11. COBOL.	SYSDEBUG (REG	SIOA).cob 🕅		
	Line 35	Column 1	Insert	5	Browse	
		-+-*A-1-B+	2+-		+5	+6
	000034	* Ini	tialize P	rogram-work-fiel	ds, Program-	flags, 🔺
2	000035	DIS	PLAY "Pro	gram REGIOA STAP	TING "	
	000036	MOV	E 2 TO BR	ANCHFLAG.		
	000037	MOV	BRRRRR	to FIELD-A.		
	000038	MOV	222222' 3	to FIELD-C.		
•	000040	MOV	E "Enterp	rise Transformat	ion POT " to	Input
	-					
	1					

164. Click the Step Over button 🗇 (or F6) twice.

Click on Variables view (Upper right) and you will then see the COBOL working-storage data items.

(×)= Vari	i 🔀 🔏 Bre 🔡 🖁 🛛	🛂 Mon	ති	Mod		
			Ľ.	⇒ti		~
Name		Value				
+ 😑	PROGRAM-WORK-FIELDS					
+ 😑	PROGRAM-FLAGS					
•	BRANCHFLAG	02				
					1	2

165. Move the mouse to the field BRANCHFLAG and wait 3 seconds or doubleclick on it. You will see the contents of this field. This is another way to see the variables

	) REGIOA.jd	参 ZUSER 11	LCOBOL.SY	SDEBUG (REGIO)	A).cob 🛛	- 6
Γ	Line 36	Colu	mn 35	Insert	Browse	
		+-*A-1-B	3+	-2+	-3+	+6
	000034	*	Init:	ialize Pro	gram-work-fields, Progr	am-flags, 🔺
	000035		DISP	LAY "Progr	am REGIOA STARTING "	
	000036		MOVE	2 TO BRAN	CRFLAG.	
۶.	000037		MOVE	AAAA	HELAG = 02 - A	
	000038		MOVE	'BBBBBB'	to FIELD-B.	
	000039		MOVE	'cccccc'	to FIELD-C.	
0	000040		MOVE	"Enterpri	se Transformation POT "	to Input-
	•					

166. Move the mouse to the line 000042, Right-click and select Jump To Location

ľ	P	DECTOR NO.		
		Line	Add Breakpoint Add Watch Breakpoint	Browse
	\$	000036 000037	Jump To Location Run To Location	FLAG.
	•	000038 000039 000040	Monitor Expression Monitor Memory	FIELD-B. FIELD-C. Transformation POT " to Input-
		000041 000042-	Switch View	<pre>PUT" to Output-name. program-to-call.</pre>
		•		

167. The program jumps from line 37 to 42 but did NOT execute the statements that were jumped. This can be verified just moving the mouse to one of the fields like FIELD-B. Also note that the breakpoint that you added did not work, since it was jumped.. See below.

	REGI0A.jcl	SUSER 11. COBOL. SY	SDEBUG(REGI0A).cob	×	- 8
	Line 3	88 Column 39	Insert	Browse	
		+-*A-1-B+	23	+	+б
	000036	MOVE	2 TO BRANCHF	LAG.	<b></b>
	000037	MOVE	'AAAAAA' to 1	FIELD-A.	
	000038	MOVE	'BBBBBBB' to	FIELD-B.	
	000039	MOVE	'CCCCCC' to		
Θ	000040	MOVE	"Enterprise	Fransformation POT "	to Input-
	000041	MOVE	"WSED - OUTPO	JT" to Output-name.	
۵.	000042	MOVE	"REGIOB" to p	program-to-call.	<b>T</b>
	•				
168. Click STEP INTO - (or F5) twice since we want to debug the Called program. After a while, you should be inside of the program REGI0B as seen below:

🎋 Debug 🛛	해 Servers	(x)= Va	ari 🛛 💊 Bre	1010 Reg	💇 Mon	ති Mod	- 0
						<u>ia</u> ⇒ti	
× ► □ ■ 	₩   🧞 • ౫. ा rm: OS/390(R) Connect	. € = Name 	RECVD-PARMS		Value		
Proces	read:1 (Runnable ) REGIOB:02 REGIOA:01 :s: 500232616 Program:				]		
REGIOA.jd	SER01.COBOL.S	YSDEBUG(REGI0B).co	b 83				
Line 2	Column 1	Insert		Bro	wse		
	+-*A-1-B+	2+3	34-	+	5	-+	6
000001	Identifica	tion Division	1.				-
• 000002	Program-ID	. "REGIOB".					
000003	******	**********	*******	*****	*****	*****	**
000004	* This pro	gram is calle	ed.				
000005	* It retur	ns a value ((	))				
000006 * Inis Will cause an error when this value is used in						.n	
000007	*******	***********	*********	*****	*****	*****	· •
•							•

169. Click STEP INTO ... (or F5) again. As you see the program REGI0B just moves 0 to In-value and returns. This value will be used in a division. And guess what? Division by Zero will later cause an exception.

	000012	US IN-VALUE PIC 99.	1
	000013		
	000014	Procedure Division using Recvd-Parms.	
>	000015	Move 0 to In-value.	
	000016	Goback.	
	000017		
	000018		÷

170. Now step over several time times until the cursor is positioned in line 46. We now want to produce a division by zero and jump over line 46 without executing line 46, which sets the RECEIVED-FROM-CALLED field again to 1. This is done only for the correctness of this program for our other labs.

So position the curser to line 47 and hit Jump to Location from the context menu.

171. Just use resume IP (F8) until you get the exception due the division by zero (0CB)

172. Select Examine and click OK since you want to fix this problem:



173. You will get a message below. Click OK

🔘 Deb	ug Engine Message	<li></li>
į)	CEE3211S The system detected a decimal-divide exception (System Completion Code=0CB). Execution suspended while examining exception from entry point REGIOA in program REGIOA at statement 47 at compile unit offset 000005BE at address 1DD00966. Application environment and data may be unstable or inconsistent.	
	OK	

- 174. Move the mouse to the line 45 (will need to scroll back) and press the left mouse button to position the cursor there.
- 175. Now Right-click on line 45 and select Jump to Location, since we want to execute the divide again. Depending on the version that you are using you might need to do that twice.
- 176. You will be now on the line 45 as shown below

	000044	CALL program-to-call USING received-from-calle
۵.	000045	move 66 to value1.
	000046	move 1 to received-from-called
	000047	divide value1 BY received-from-called GIVING 2
	000040	DTODINY HTLE

177. Now step over twice (F6), to get into line 47.

178. Using the Variables view, expand PROGRAM-WORK-FIELDS, navigate to RECEIVED-FROM-CALLED field and click on its value 01, since we executed the Move in line 46.

The value will become editable. You can change it to a desirable number; you also can produce a division by zero.

🗱 = Variables 🕴 🔍 💁 Breakpoint	s 👯 Registers 🖾 Monitors 🕺 Modules
Name	Value
🖂 🔍 PROGRAM-WORK-FIELDS	
INPUT-NAME	'Enterprise Transformation POT '
OUTPUT-NAME	'WSED - OUTPUT '
PROGRAM-TO-CALL	'REGIOB '
🔍 RECEIVED-FROM-CALLE	01
VALUE1	66
1	
	😥= Variables 🙁 💁 Breakpoints 🚻 Registers 💷 Monitors 🖞 Modules
	Name Value
	🖃 🛡 PROGRAM-WORK-FIELDS
	INPUT-NAME 'Enterprise Transformation POT '
	OUTPUT-NAME 'WSED - OUTPUT '
	PROGRAM-TO-CALL 'REGIOB '
	📃 🧠 RECEIVED-FROM-CALLEE 🔟 🚩
	VALUE1 66

- 179. Depending of your modification, you might now correct it a second time, like described in the last steps. If you have no division by zero, proceed with the next step.
- 180. Click STEP INTO 3 (or F5) until you see the next called program REGI0C being called
- 181. Click on Step Over 😒 (F6) and 🕪 (F8) when you are satisfied so the program will execute until the end and click OK.

💽 Debug Engine Message	×
Application has Terminated	
	OK

182. Using the Remote Systems view locate the node demomvs and expand the JES node and either the general filter "My Jobs" or the filter "generated" which has been created in a previous step to specifically display your GEN\* jobs.

183. You will see the job that you had submitted as the example below. If your job does not appear in the list, click on the filter and press F5 for a refresh.



184. Expand the job clicking on the + sign and double click in the step GO, you can also see all the steps of the submitted job



- 185. Close the editor (CTRL +F4)
- 186. Select the all your jobs (use CTRL key), right-click and select Purge. All job listing will be purged if you have authorization for.

## 3.8 Copybook expansion and dependency check

- 187. Add IGYTSALE.cbl to your batch Cobol project (this has been copied to your Cobol dataset) and open it from there.
- 188. With the editing window of IGYTSALE.cbl active, press Ctrl +F to bring up the Find/Replace dialog (shown below). Enter copy in the Find field and click the All button.

Eind COPY	Next Pr	e <u>v</u> ious All
Replace	<u>R</u> eplace	Replace all
Case sensitive Whole word Regular expression V Wrap Select four	nd text Pe	eek: 0 🛨 🗖
Restrict search to selection 🔲 Restrict search to columns Start column 🚺 E	nd colu <u>m</u> n	80

189. You should see something similar to the following. All lines with copy will be shown.

	IG	TSALE.cbl		, 🗆
	I	ine 112	12 Column 31 Insert	
			+-*A-1-B+2+3	-6
	θ			▲
⇒	_	000112	<pre>2 *** Include code: (from copy library)</pre>	
E.	H	000113	3 *** IGYTCRC - Copy member in IGY.V1R2M0.SIGYSAMP	
E.	H	000115	5 *** IGYTPRC - Copy member in IGY.V1R2M0.SIGYSAMP	
R	H	000117	7 *** IGYTSRC - Copy member in IGY.V1R2M0.SIGYSAMP	
2	H	000166	6 *** 97Feb Show where copy files are located on	
K	H	000216	6 ISOSOO Copy igytere.	
Ľ.	H	000222	2 Copy igytprc.	
Ľ,	B	000226	s copy igytard.	
Ē.	8	003036	Copy igytere.	-
⇒	Ц	003050	0 Copy igytpro.	
⇒	Ä	003360	0 Copy igytore.	
⇒	Ä	003365	5 Copy igytere.	
	<u> </u>			-
	4		<u> </u>	
	Ei	nd COPY	( <u>N</u> ext Pre <u>v</u> ious All	
	R	eplace	Replace Replace all	ī
		cElace 1		┛╴
	Γ	<u>C</u> ase ser	ensitive 🔲 Whole word 🔲 Regular expression 🔽 <u>W</u> rap 🔲 Select found text Peek: 🚺 🏯	
	Γ	Res <u>t</u> rict s	t search to selection 🔲 Restrict search to columns Start column 1 End column 80	A.

190. Highlight the copybook name igytcrc (line 216), press mouse button 2, and select the Open Copy Member action.

$\sim$	U.	000100	***	g/rep	Snow	wnere	сору	Illes	are	Tocated	on	٧M	GENAS LTA * *	
⇒	θ	0002161	IS0500	Сору	igytcrc.									
⇒	8	000222		Сору	igytprc.									
$\Rightarrow$	Г.	000228		Conv	invterc									

191. A window will open and ask if you want to add this Data Set Member to your Project. Click No



192. This will open the igytcrc copybook in another editing window as shown below:

IGYTSALE.cbl	📄 IGYTCRC.cpy δ	3		- 8
Line 1	Column 1	Insert		
-	+-*A-1-B+	2+3	+5+	6
000001	15	cust-record-type	Pic x.	
000002	15	cust-number	Pic xx.	
000003	15	cust-name	Pic x(20).	
000002 000003	15 15	cust-number cust-name	Pic xx. Fic x(20).	

Note that the copybook is shown even that is NOT part our MVS Project. Usually there is no sense to add copybook libraries to MVS projects, since they are used for many projects, unless they also need updates.

Copybooks are resolved based on the value of Copy Libraries in the COBOL Settings for the MVS project. The COBOL Settings of the property group of your remoteCOBOL MVS project should look like the following figure:

Cobol Compile Step Option	15 X
Compile Procedure Name:	
ELAXFCOC	
Compile Procedure Step Name:	
COBOL	
Compiler Options:	
Listing Output Data Set:	
Debug Data Set:	
<hlq>.COBOL.SYSDEBUG</hlq>	
Object Deck Data Set:	
<hlq>.COBOL.OBJ</hlq>	
Copy Libraries:	[]
<hlq>.COBOL.COPYLIB</hlq>	Data set to search for copybooks

In this example, the copybook will be resolved (expanded) from USER##.COBOL.COPYLIB. You can specify multiple copy libraries, each separated by a space. This is equivalent to what on MVS is termed concatenation, in effect specifying the search order for copybook resolution/expansion.

193. Close the IGYTCRC.cpy editor.

194. Click anywhere on the IGYTSALE.cbl editor area to make it the active editor. Right click and select the Show all action.

$\widehat{\Upsilon}$	Ð	000112 000113	***	Include coo IGYTCRC	de: (from copy lik - Copy member in	IGY.V1R2M0.SIGYSAMP	1
2	븮	000115	**	Save		IGY.V1R2MO.SIGYSAMP	
⇒	A	0001166	**	Cut	Ctrl+X	iles are located on	
⇒	Ħ	000216	IS0500	Сору	Ctrl+Insert		
⇒	8	000222		Paste	⊂trl+∀		
⇒	8	000228	-			-	
⇒	8	003036		Select	•		
⇒	Θ	003043		Selected	•		
⇒	Θ	003050		Deselect	Alt+U		
⇒	Η	003360		Filter view			
⇒	Ð	003365		Show all	CHIAW		
	4	003370		SHOW all	Cultw		

This action removes the filtered view (currently showing all statements with the word copy) and displays all statements.

RDz has a feature named Show dependencies that is useful when you need to identify which components are required to compile the COBOL program. For example when working off-line you will need to identify all dependencies that are necessary for a clean compilation.

195. Right-click on IGYTSALE.cbl in the z/OS Projects view and select Show Dependencies.

196. The COBOL compiler will be used to find the dependencies. You will see that this program will need all the members listed below.

Show Dependencies The following is the list of build dependencies	
The build dependencies for 'remoteCobol'	
ZUSER11.COBOL(IGYTSALE) ZUSER11.COBOL.COPYLIB(IGYTCRC) ZUSER11.COBOL.COPYLIB(IGYTPRC) ZUSER11.COBOL.COPYLIB(IGYTSRC)	
Select All Deselect All	
Add selected to subproject	
•	Finish Cancel

197. Click on Finish to close this dialog. (Do not select any member)