

Developing and Debugging CICS Programs Using Rational Developer for System z (RDz)

Pradeep Gohil
CICS Software Engineer
gohilpr@uk.ibm.com



IBM CICS® User Conference 2009

Abstract

With the release of IBM Rational Developer for System z V7.5 (RDz) it is now easier than ever to integrate the development of CICS application programs with a single user interface. Whatever your language of choice (COBOL, PL/I, C, C++, Assembler or Java) RDz allows you to code, compile and debug your programs directly from the mainframe.

This presentation focuses on the traditional CICS programming languages (COBOL, PL/I, C, C++, Assembler) and demonstrates the features integrated into RDz for software development on the mainframe. A CICS application is taken through the steps of coding, compilation and run-time debugging as part of a live rolling demo. A variety of productivity enhancement features in RDz are demonstrated along the way. The presentation material provides a documented reference for the configuration steps required (both in RDz and the mainframe) to set up each stage of the application development process.

IBM CICS® User Conference 2009

© IBM Corporation 2009. All Rights Reserved.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries: ibm.com/legal/copytrade.shtmlAIX, CICS, CICSplex, DataPower, DB2, DB2 Universal Database, i5/OS, IBM, the IBM logo, IMS/ESA, Power Systems, Lotus, OMEGAMON, OS/390, Parallel Sysplex, pureXML, Rational, Redbooks, Sametime, SMART SOA, System z, Tivoli, WebSphere, and z/OS.

A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office

Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Agenda

- Introduce Rational Developer for System z (RDz)
- Configuring Debug Tool for CICS
- Live Demo!
- Benefits of RDz over ISPF programming

Rational Developer for System z (RDz)

- Eclipse based Integrated Development Environment (IDE)
- One stop portal for coding and deploying CICS applications
 - Supports COBOL, PL/I, C, Assembler, Java
- Remotely manage mainframe resources
MVS datasets, USS files
- Remotely debug CICS applications
- Deploy CICS applications as Web services
- Generate CICS code for invoking Web services
- See for yourself....

Configuring CICS to use the Debug Tool

- Follow the steps:
 1. Add Debug Tool library to CICS JCL
 2. Create debugging profiles data sets
 3. Upgrade CSD with debug definition
 4. Configure EQA group to install on CICS startup
 5. Enable debug mode in CICS

Add Debug Tool library to CICS JCL

- Add the Debug Tool SEQAMOD data set to DFHRPL
 - DSN=PP.DEBUG.V910.SEQAMOD
- Ensure LE run-time libraries are in DFHRPL
 - DSN=CEE.SCEECICS
 - DSN=CEE.SCEERUN

Create debugging profiles data sets

- Use IDCAMS to create and initialize the VSAM data sets
 - DFHDPFMB – base data set
 - DFHDPFMP – path data set
 - DFHDPFMX – alternate index data set
- Create file definitions for the data sets
 - Sample JCL in CICS TS 3.2 InfoCenter for:
 - VSAM RLS
 - VSAM non-RLS
 - Remote files
- File resources need to be installed at CICS initialization
 - Add to a group in a group list

Upgrade CSD with debug definitions

- Locate member EQACCCSD in Debug Tool's SEQASAMP data set
- Follow instructions in EQACCCSD and uncomment definitions as necessary
- Upgrade CSD with new definitions - creates Group(EQA)

Configure EQA group to install on CICS startup

- Group(EQA) must be installed at CICS bring up so add to a group list
 - Add Group(EQA) to a group list
 - `CEDA ADD GROUP(EQA) LIST(MYLIST)`
 - Add group list to GRPLIST SIT parameter
 - `GRPLIST=(*FHLIST, MYLIST),`

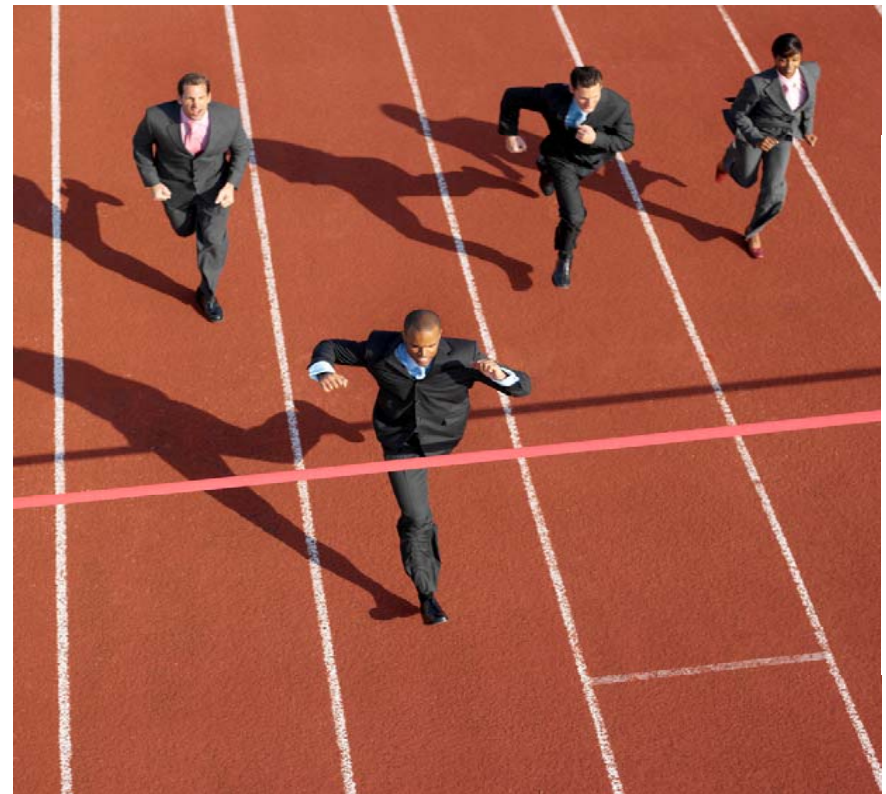
Enable debug mode in CICS

- Specify as a SIT parameter
 - `DEBUGTOOL=YES`
- Or enable in run-time using
 - `CEMT SET SYSTEM DEBUG`

Configuring programs to use the Debug Tool

- Include the TEST option in compiler parameters
 - **PL/I**
 - PARM(..., TEST)
 - **COBOL**
 - PARM(..., TEST(ALL, SEPARATE))
 - //SYSDEBUG DD DISP=OLD,DSN=SCOTTC.SYSDEBUG(SWITCH)
 - **C**
 - PARM(TEST)
 - Source from translation step must be saved and passed to the compiler
- Locations of source files are stored in the compiled modules

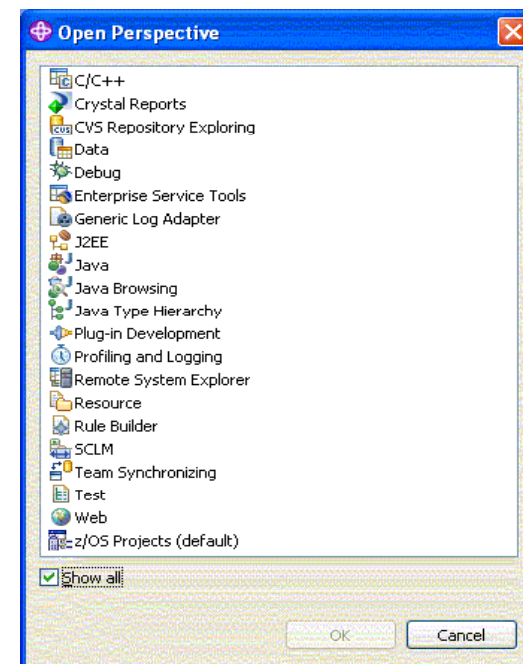
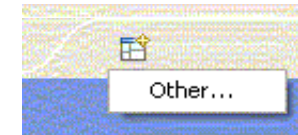
Live demo!



- The following pages are for reference purposes

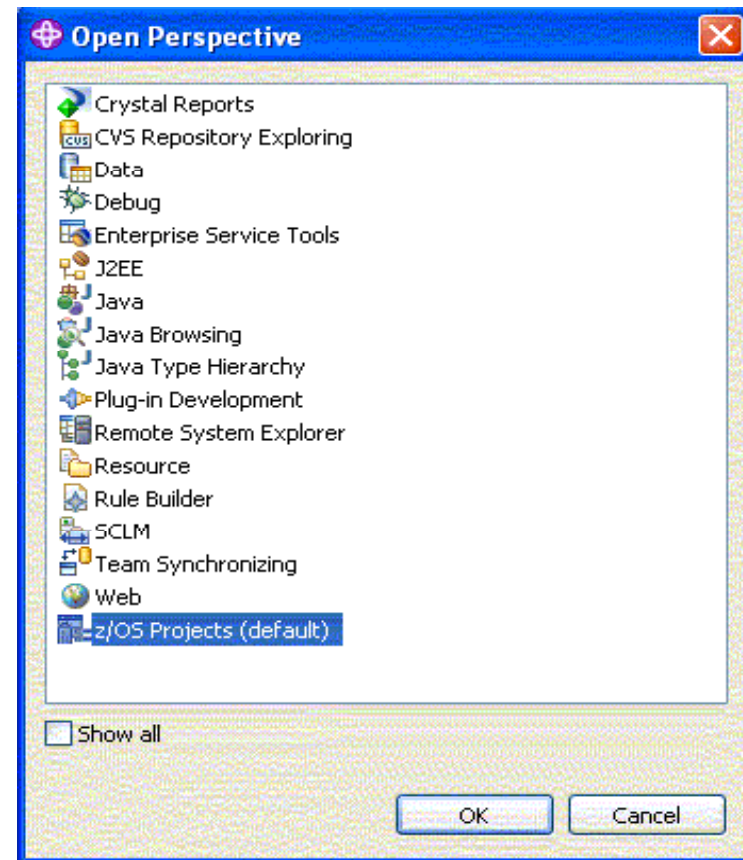
Perspectives

- Click on open perspective button
- Select a perspective from the list

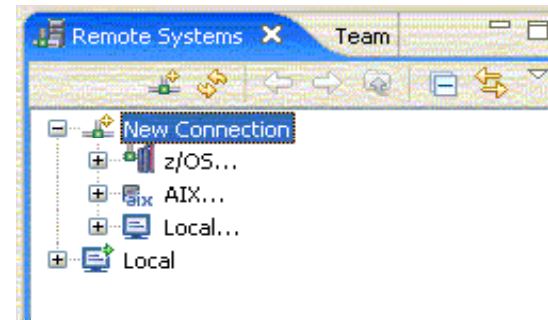


z/OS Projects

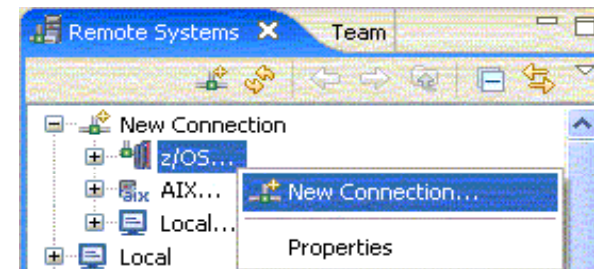
- Select the z/OS Projects perspective



- Remote systems panel opens up



- Right click on z/OS and create a new connection



- Enter connection details and press next

New Remote System Connection
Define connection information

Parent profile: Hursley

Connection name: winmvs26

System type: z/OS

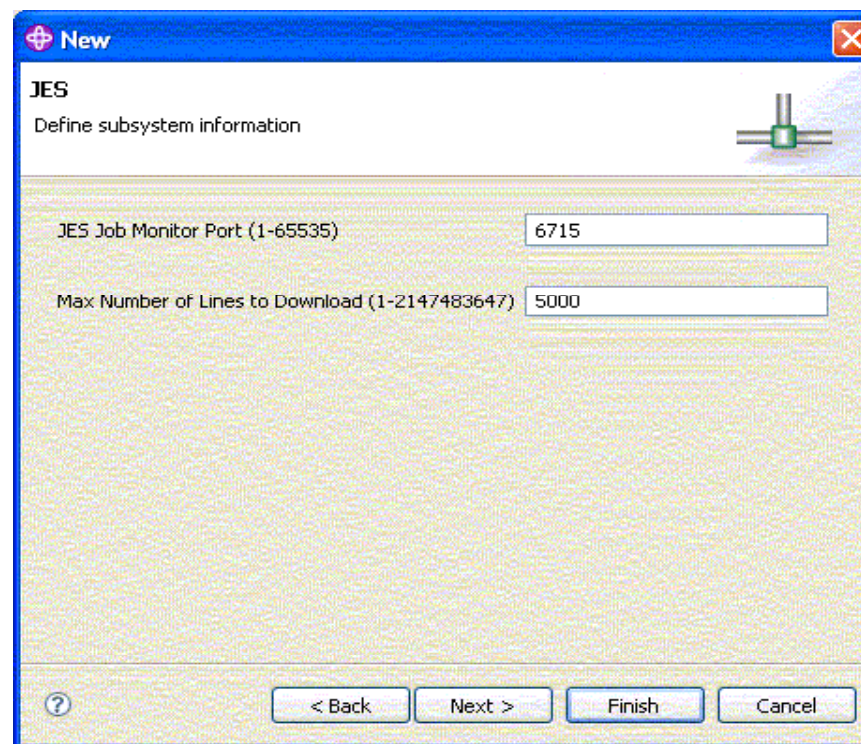
Host name: winmvs26.hursley.ibm.com

Description:

Verify host name

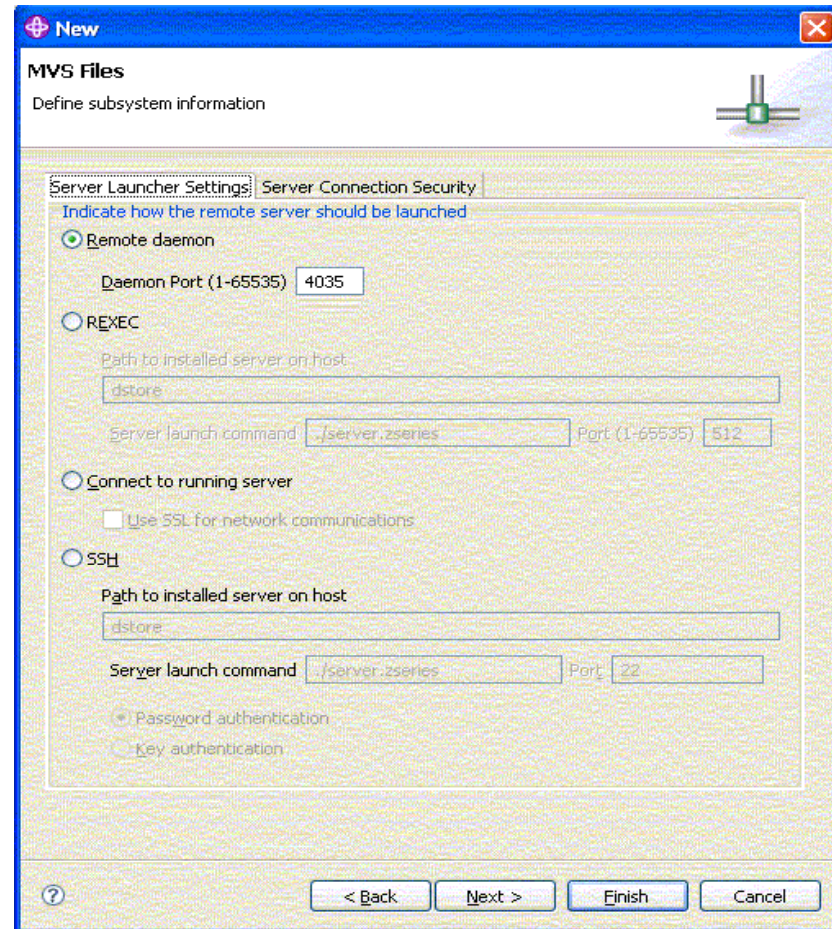
? < Back Next > Finish Cancel

- Enter port numbers

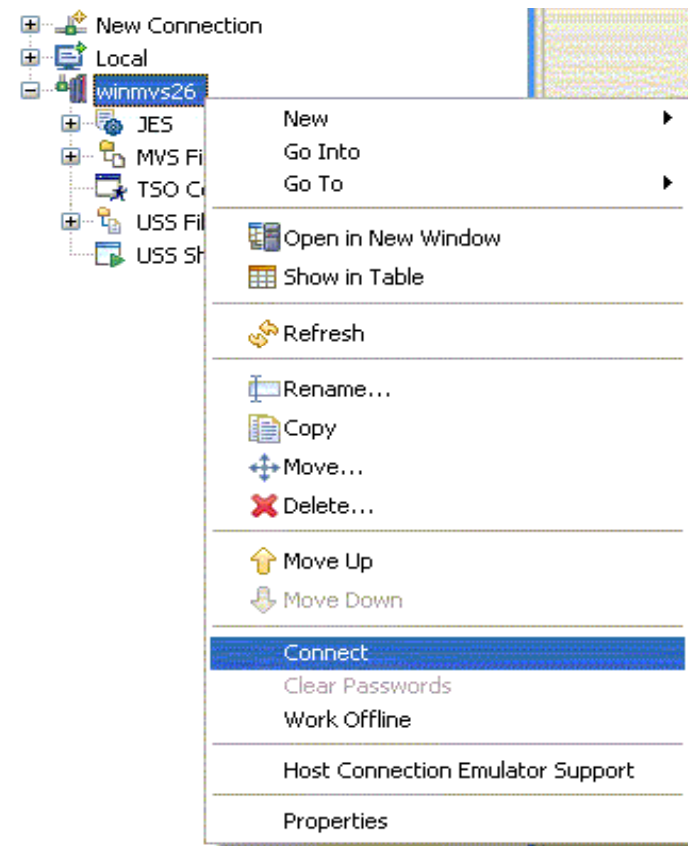


The screenshot shows a Windows-style dialog box titled "New" with a blue header bar. Below the header, the text "JES" is displayed in bold, followed by "Define subsystem information" and a small icon of a green plug. The main area contains two text input fields. The first field is labeled "JES Job Monitor Port (1-65535)" and contains the value "6715". The second field is labeled "Max Number of Lines to Download (1-2147483647)" and contains the value "5000". At the bottom of the dialog, there is a help icon (question mark) on the left and four buttons: "< Back", "Next >", "Finish", and "Cancel".

- Select connection type



- Right click on system icon and connect



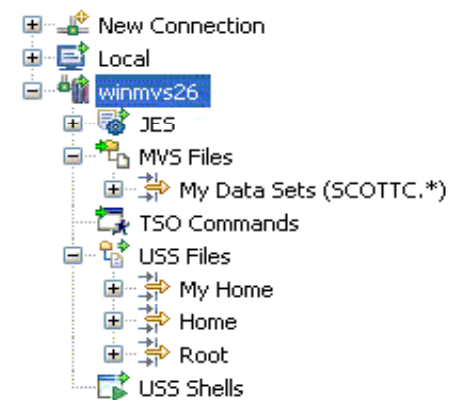
- Input username and password for system (TSO logon credentials)



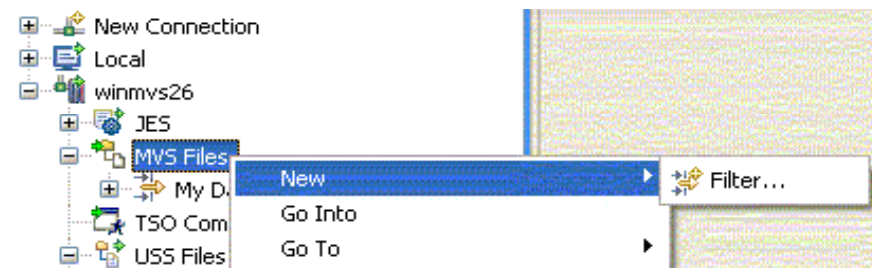
Connecting to WINMV526.HURSLEY.IBM.COM Initializing Remote Systems Explorer host server

- Observe connecting status

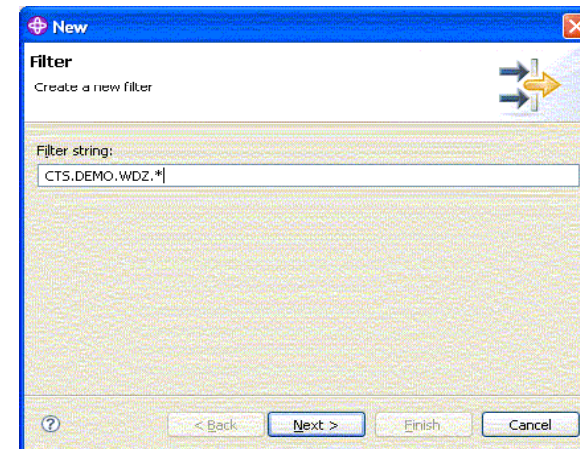
- Once connected expand the MVS Files and USS Files sections



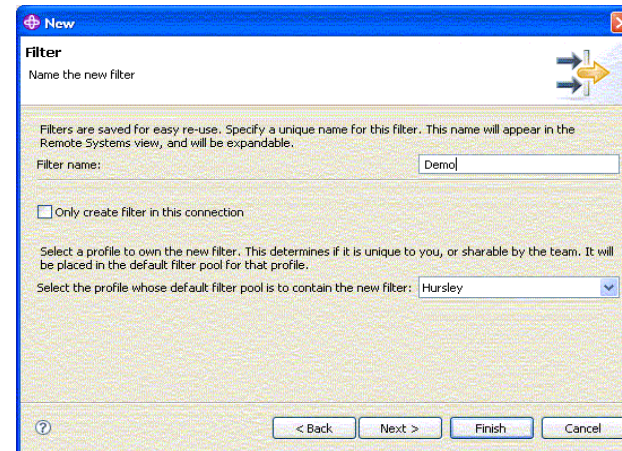
- Right click on MVS Files and create new filter



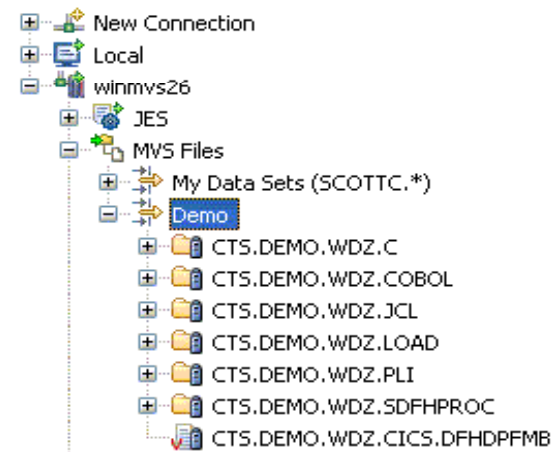
- Input filter criteria



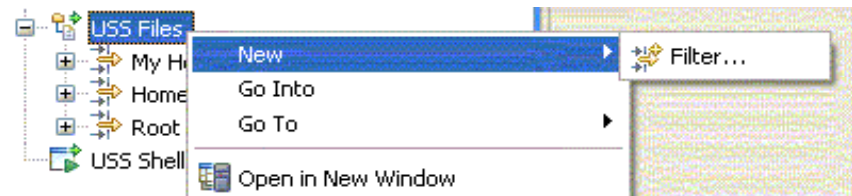
- Specify a name for the filter



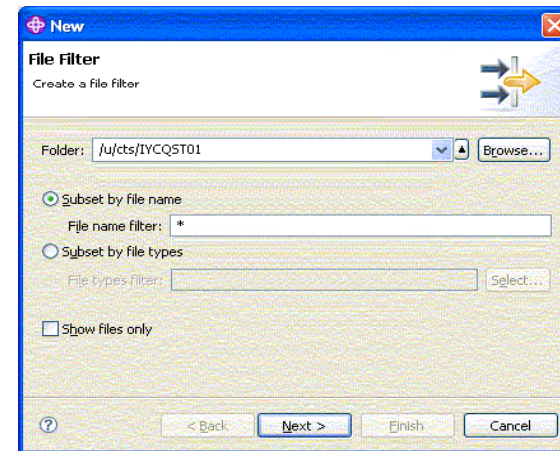
- Expand the filter icon to see data sets



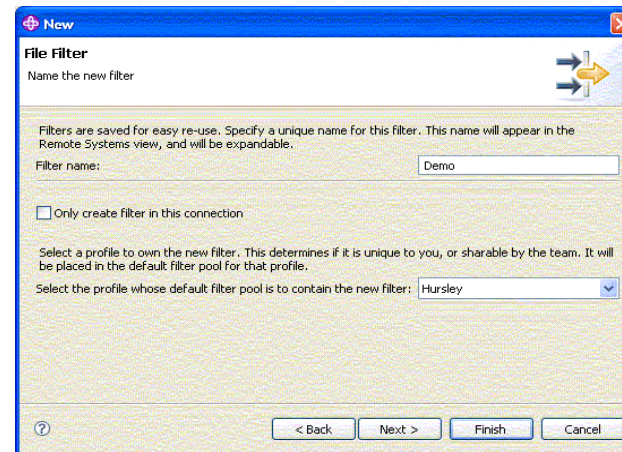
- Right click on USS Files and create new filter



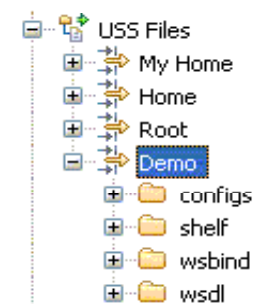
- Input filter criteria



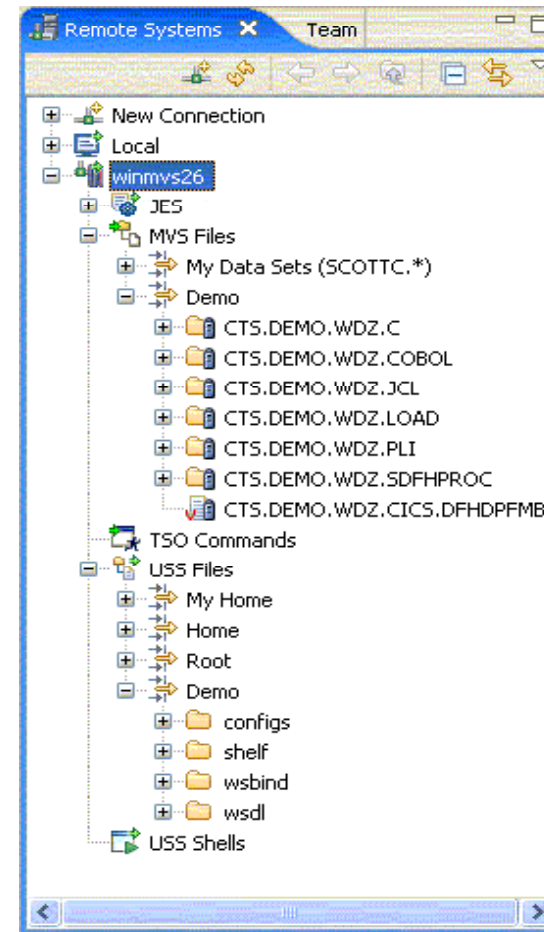
- Specify a name for the filter



- Expand the filter icon to see data sets



- Now have connection with filters in place



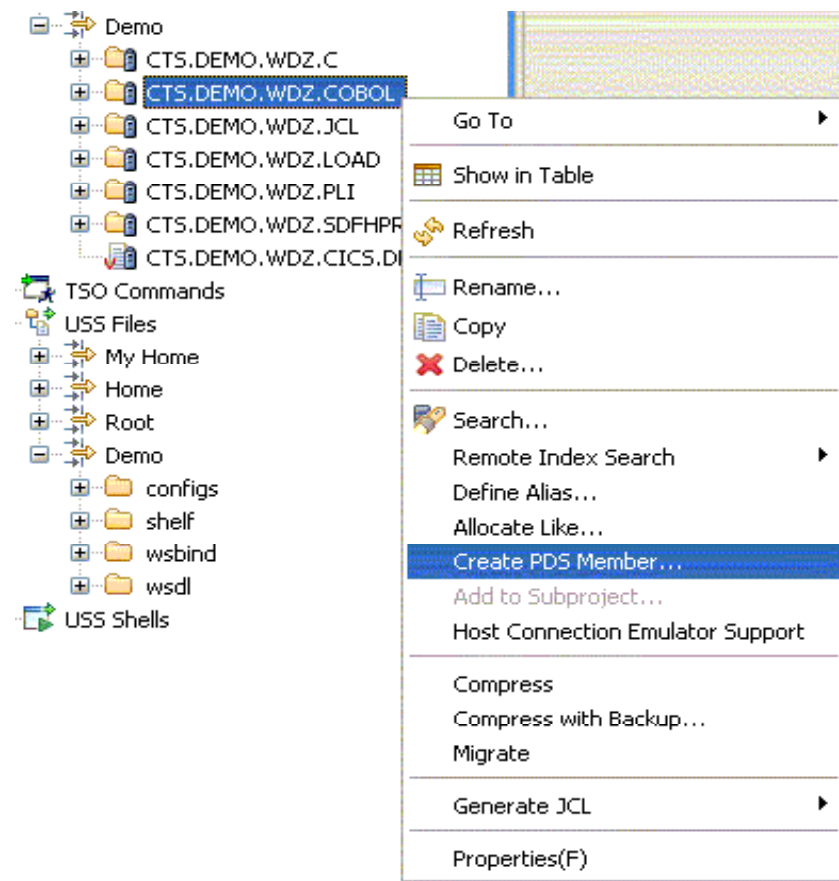
- Observe file system mappings window
 - Includes code page information

The screenshot shows a window titled "z/OS File System Mapping" with a dropdown menu set to "winmvs26". Below the menu is a table with five columns: Mapping Criterion, Workstation File Extension, Transfer Mode, Host Code Page, and Local Code Page. The table lists various file types and their corresponding mappings.

Mapping Criterion	Workstation File Extension	Transfer Mode	Host Code Page	Local Code Page
**COBOL	cbl	text	IBM-037 (inherited)	Cp1252 (inherited)
**COBCOPY	cpy	text	IBM-037 (inherited)	Cp1252 (inherited)
**PLI	pli	text	IBM-037 (inherited)	Cp1252 (inherited)
**ASSEMBLE	asm	text	IBM-037 (inherited)	Cp1252 (inherited)
**OBJ	obj	binary	IBM-037 (inherited)	Cp1252 (inherited)
**LOAD	exe	binary	IBM-037 (inherited)	Cp1252 (inherited)
**CLIST	cmd	text	IBM-037 (inherited)	Cp1252 (inherited)
**JCL	jcl	text	IBM-037 (inherited)	Cp1252 (inherited)
**SIGYCLST	cmd	text	IBM-037 (inherited)	Cp1252 (inherited)
**CNTL	jcl	text	IBM-037 (inherited)	Cp1252 (inherited)
**LISTING	lst	text	IBM-037 (inherited)	Cp1252 (inherited)
**OUTLIST	out	text	IBM-037 (inherited)	Cp1252 (inherited)
**INCLUDE	inc	text	IBM-037 (inherited)	Cp1252 (inherited)
**MACRO	mac	text	IBM-037 (inherited)	Cp1252 (inherited)
ERRWDZ	err	binary	UTF-16BE	UTF-16BE
**COPYLIB	cpy	text	IBM-037 (inherited)	Cp1252 (inherited)
**XML	xml	text	IBM-037 (inherited)	Cp1252 (inherited)
**BMS	bms	text	IBM-037 (inherited)	Cp1252 (inherited)
**C	c	text	IBM-037 (inherited)	Cp1252 (inherited)
**CPP	cpp	text	IBM-037 (inherited)	Cp1252 (inherited)
**H	h	text	IBM-037 (inherited)	Cp1252 (inherited)
**HPP	hpp	text	IBM-037 (inherited)	Cp1252 (inherited)
**MFS	mfs	text	IBM-037 (inherited)	Cp1252 (inherited)

Writing a CICS COBOL application

- Right click on data set and create a PDS member



- Give the member a name

New PDS Member

Create PDS Member
Create a new PDS member residing on z/OS.

Data Set: CTS.DEMO.WDZ.COBOL

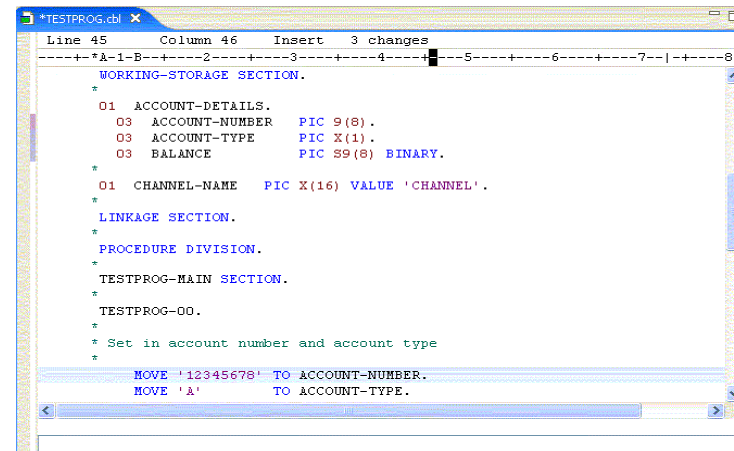
Member Name: TESTPROG

Finish Cancel

- Double click on TESTPROG to see empty file in editor

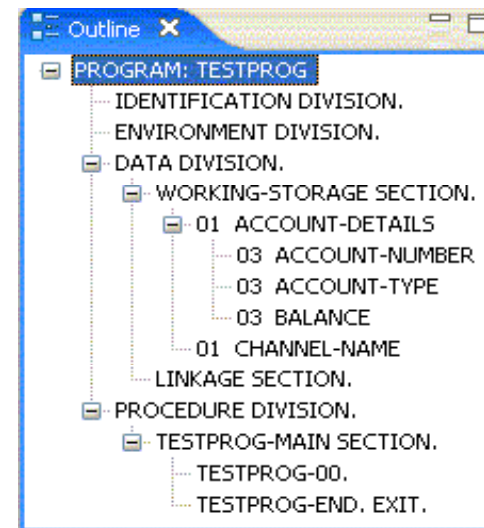


- Type in COBOL source code

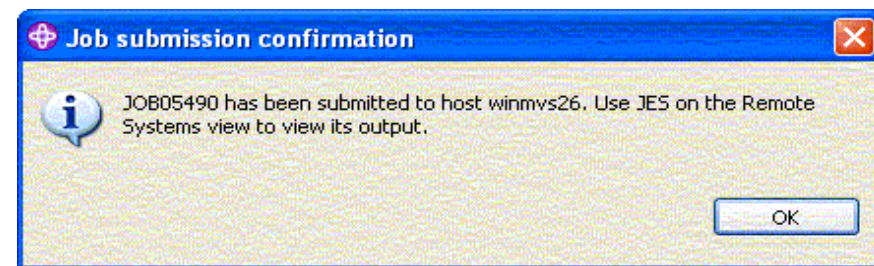
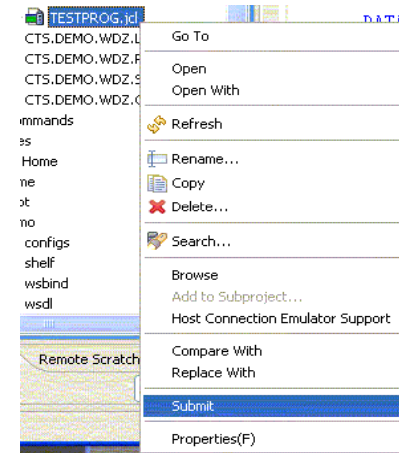
A screenshot of a COBOL editor window titled '*TESTPROG.cbl'. The window shows the source code for the program. The status bar indicates 'Insert 3 changes'. The code is as follows:

```
-----*A-1-B-----2-----3-----4-----5-----6-----7--|--8
WORKING-STORAGE SECTION.
*
  01 ACCOUNT-DETAILS.
     03 ACCOUNT-NUMBER   PIC 9(8).
     03 ACCOUNT-TYPE    PIC X(1).
     03 BALANCE          PIC S9(8) BINARY.
*
  01 CHANNEL-NAME      PIC X(16) VALUE 'CHANNEL'.
*
LINKAGE SECTION.
*
PROCEDURE DIVISION.
*
TESTPROG-MAIN SECTION.
*
TESTPROG-00.
*
* Set in account number and account type
*
MOVE '12345678' TO ACCOUNT-NUMBER.
MOVE 'A'       TO ACCOUNT-TYPE.
```

- Observe program window

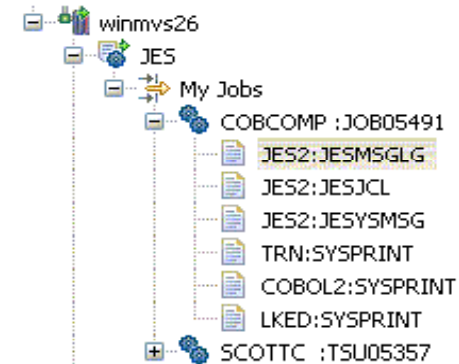


- Right click on compilation JCL and submit



- Click OK on submission confirmation

- View job logs



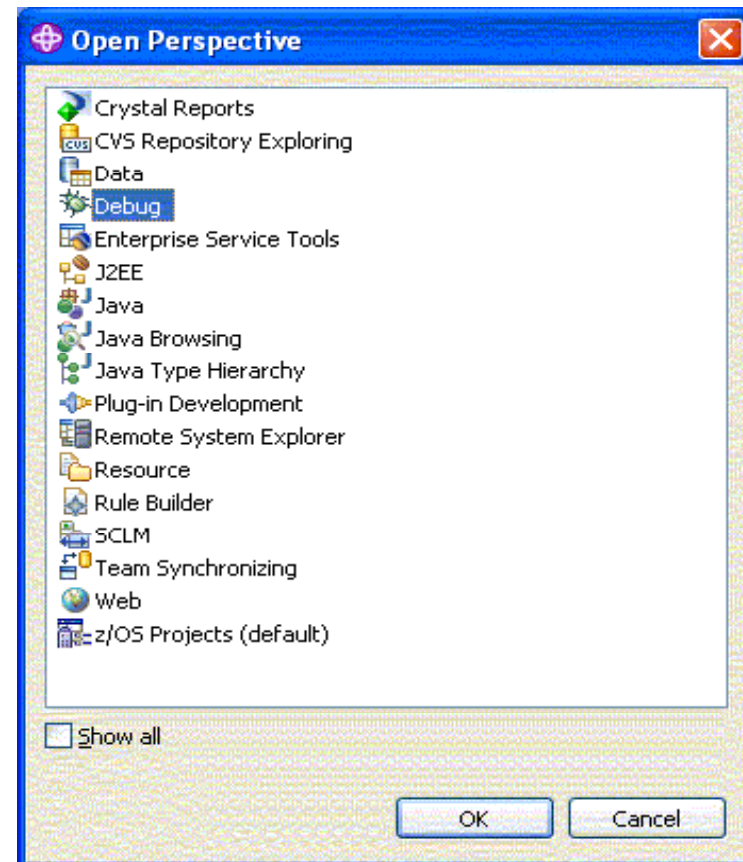
- Double click to see log information

```

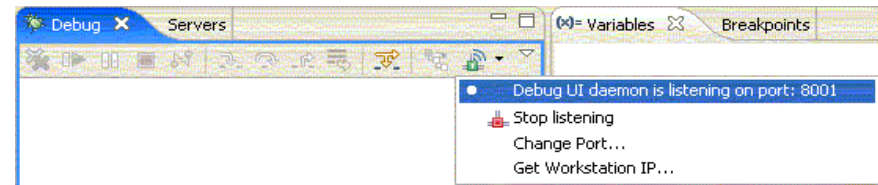
TESTPROG.cbl  SCOTTC.COBCOMP.JOB05491.D0000002.JESMSG LG
-----
JES2 JOB LOG -- SYSTEM MV26 -- NODE
-----
13.54.11 JOB05491 ---- SATURDAY, 28 JUL 2007 ----
13.54.11 JOB05491 IRRO10I USERID SCOTTC IS ASSIGNED TO THIS JOB.
13.54.11 JOB05491 ICH70001I SCOTTC LAST ACCESS AT 13:51:17 ON SATURDAY, JULY 2
13.54.11 JOB05491 $HASP373 COBCOMP STARTED - INIT 13 - CLASS A - SYS MV26
13.54.11 JOB05491 IEF403I COBCOMP - STARTED
13.54.11 JOB05491 -
13.54.11 JOB05491 --JOBNAME STEPNAME PROCSTEP RC EXCP CPU SRB CLOCK
13.54.11 JOB05491 --COBCOMP TRN 00 51 .00 .00 .00
13.54.12 JOB05491 --COBCOMP COBOL2 00 283 .00 .00 .00
13.54.12 JOB05491 --COBCOMP IEBCGENER 00 7 .00 .00 .00
13.54.12 JOB05491 --COBCOMP LKED 00 153 .00 .00 .00
13.54.12 JOB05491 IEF404I COBCOMP - ENDED
13.54.12 JOB05491 --COBCOMP ENDED. NAME- TOTAL CPU TIME=
13.54.12 JOB05491 $HASP395 COBCOMP ENDED
----- JES2 JOB STATISTICS -----
28 JUL 2007 JOB EXECUTION DATE
89 CARDS READ
1,377 SYSOUT PRINT RECORDS
0 SYSOUT PUNCH RECORDS
89 SYSOUT SPOOL KBYTES
0.01 MINUTES EXECUTION TIME
    
```

Debugging a COBOL program

- Open up Debug perspective



- Click on debug daemon icon and make note of the port it is listening on



- Go to CADP Web interface and create a new debug profile

The screenshot displays the IBM CICS Transaction Server for z/OS CADP Web interface. The user 'SCOTTCC' is signed on to the application 'IYCQST01'. The interface is divided into a left-hand navigation menu and a main content area.

Navigation Menu:

- Create compiled profile
- Create Java profile
- Create EJB profile
- Create CORBA profile
- Edit compiled profile**
 - List all profiles
 - List compiled profile details
 - List Java profile details
 - List EJB profile details
 - List CORBA profile details
- Help

Main Content Area:

Debugging profile TEST owned by SCOTTCC activated by SCOTTCC to 9.146.175.75

CICS resources to debug (use * to specify generic values, e.g. *, A*, AB*, etc)

Transaction * Applid IYCQST01
Program TESTPROG Userid *
Compile unit * Termid *
Netname *

Debug Tool Language Environment Options

Test level ALL
Command file
Prompt level PROMPT
Preference file

Other Language Environment Options

Buttons: Create Replace Save options as default

- Activate profile
 - Set session type to TCP and input IP address of machine running RDz
 - Match port number with that displayed on RDz debug daemon

The screenshot shows the IBM CICS Transaction Server for z/OS interface. The user is logged in as SCOTTTC to the application IYCQST01. The screen displays the 'Compiled language debugging display device' configuration options. The 'TCP/IP' option is selected, with the IP address set to 9.146.164.95 and the port to 08001. The 'Type of socket communication' is set to 'Single'. The '3270 display terminal' option is unselected. There is a checkbox for 'In the future, do not show this page when activating profiles' which is also unselected. The interface includes a 'Save and continue' button and a 'Cancel' button at the bottom.

IBM CICS® Transaction Server for z/OS™
SCOTTTC signed on to applid IYCQST01 Help

Compiled language debugging display device

TCP/IP address or name 9.146.164.95
Port 08001
Type of socket communication Single

3270 display terminal TC26

In the future, do not show this page when activating profiles

Set compiled debugging display device (checked at profile activation time)
Help

Save and continue Cancel

- Profile is active and ready to debug program

SC01TC signed on to applid IYCQST01

1 debugging profile(s) were activated.

Only show profiles created by my userid

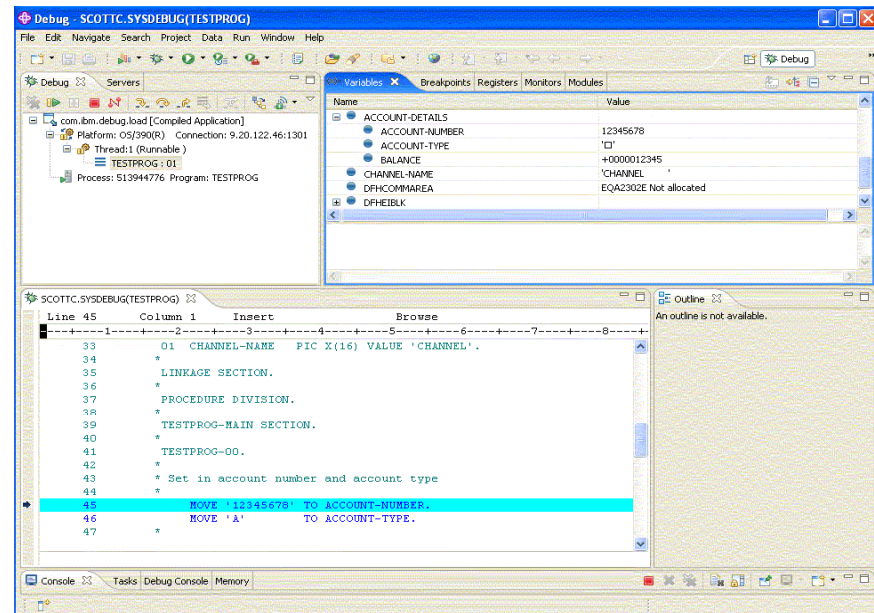
Only display active profiles

For debugging profiles which you own, click on the profile name to edit the profile.
For other users' debugging profiles, click on the profile name to view the profile.

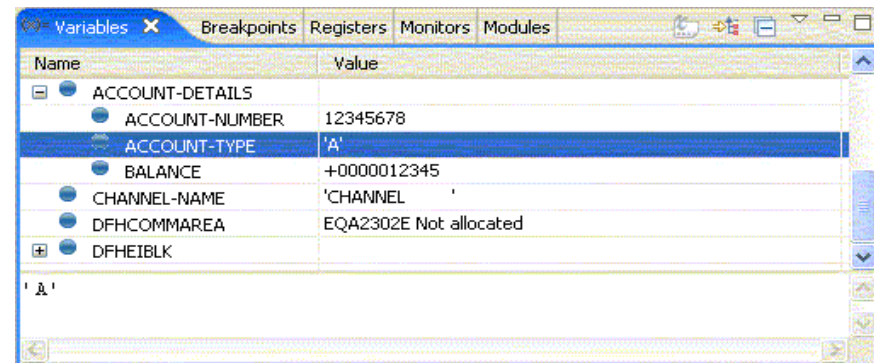
Owner	Profile	Status	Tranid	Applid	Userid	Type
<input type="checkbox"/> SC01TC	TEST	Act	*	IYCQST01	*	Compiled

Activate Inactivate Copy Delete Select all Deselect all Refresh

- As transaction starts it connects to RDz and debugger shows TESTPROG source code



- As variables are created their values appear in window

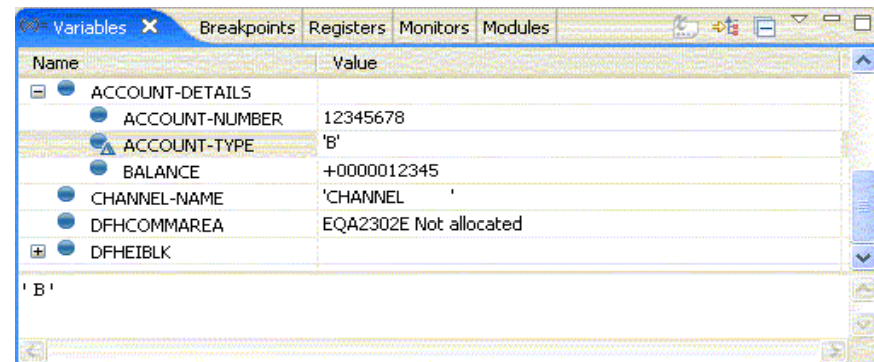


The screenshot shows the 'Variables' window in IBM CICS. The window has tabs for 'Variables', 'Breakpoints', 'Registers', 'Monitors', and 'Modules'. The 'Variables' tab is active, displaying a table with two columns: 'Name' and 'Value'. The table contains the following data:

Name	Value
ACCOUNT-DETAILS	
ACCOUNT-NUMBER	12345678
ACCOUNT-TYPE	'A'
BALANCE	+0000012345
CHANNEL-NAME	'CHANNEL '
DFHCOMMAREA	EQA2302E Not allocated
DFHEIBLK	

Below the table, there is a text area containing the character 'A'.

- Values can be modified in window and will be reflected in the running CICS program



The screenshot shows the 'Variables' window in IBM CICS, similar to the first screenshot. The 'Variables' tab is active, displaying a table with two columns: 'Name' and 'Value'. The table contains the following data:

Name	Value
ACCOUNT-DETAILS	
ACCOUNT-NUMBER	12345678
ACCOUNT-TYPE	'B'
BALANCE	+0000012345
CHANNEL-NAME	'CHANNEL '
DFHCOMMAREA	EQA2302E Not allocated
DFHEIBLK	

Below the table, there is a text area containing the character 'B'.

- Right click next to a line of source code to add a breakpoint

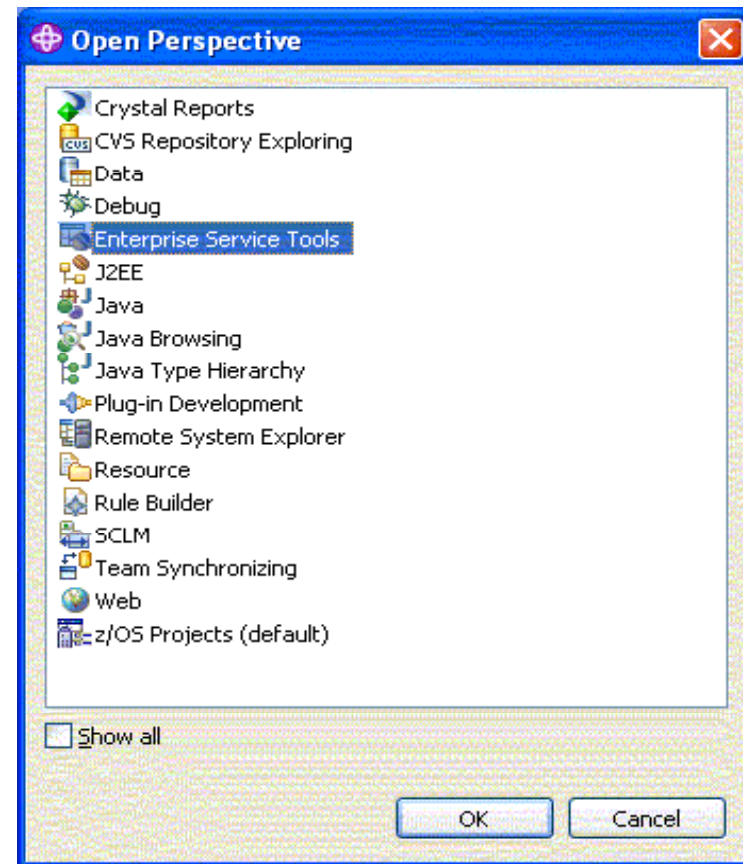


- Or choose Run To Location

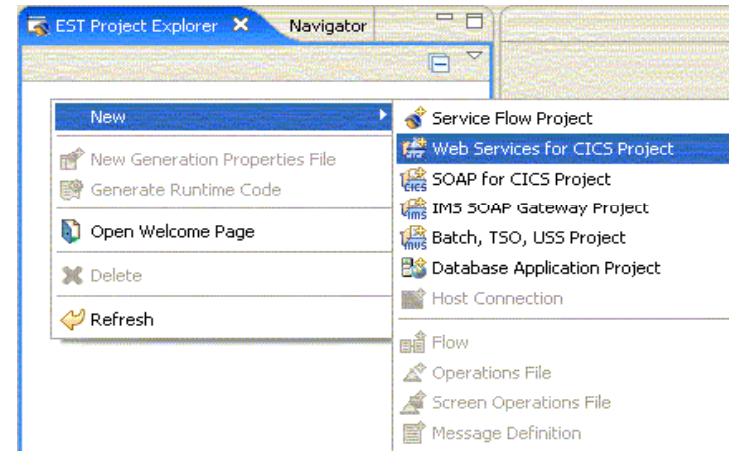


Expose as Web service

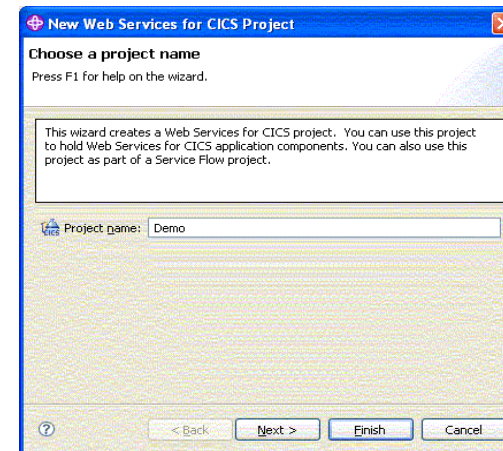
- Open Enterprise Service Tools perspective



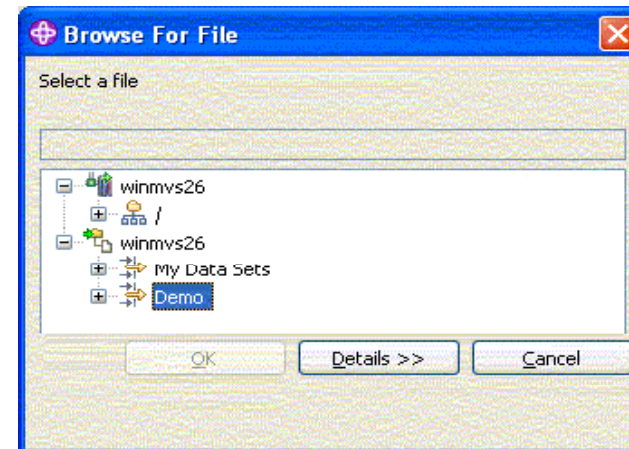
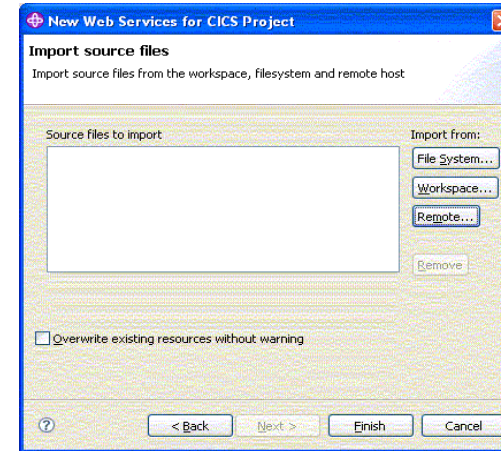
- Right click in EST Project Explorer window and create new Web Services for CICS Project



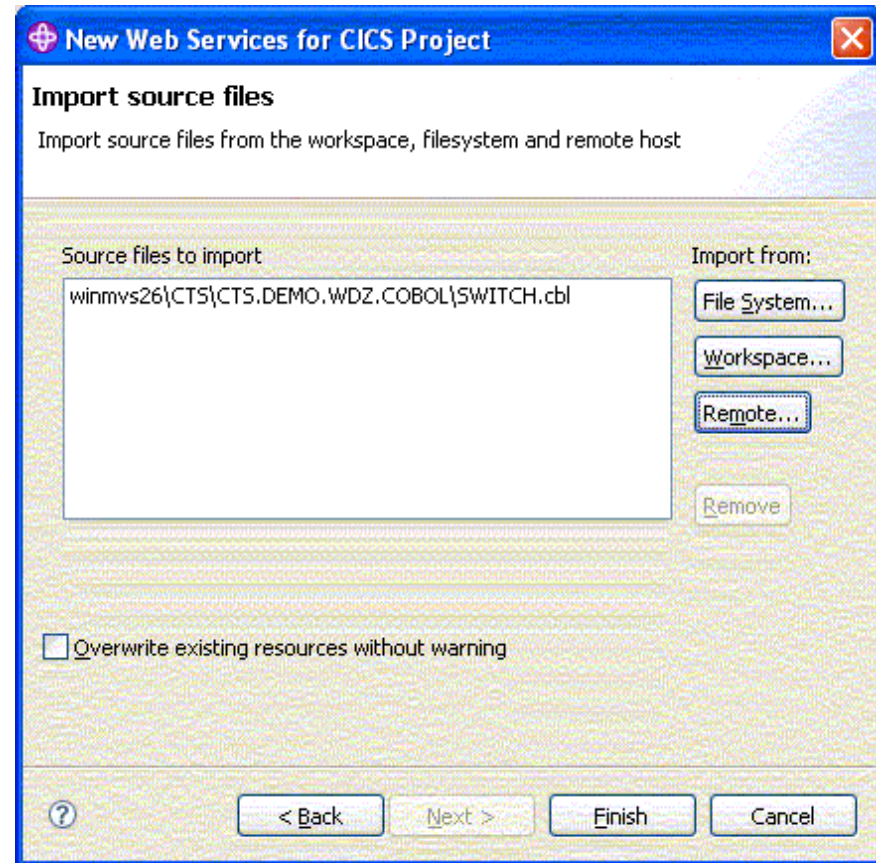
- Give the project a name



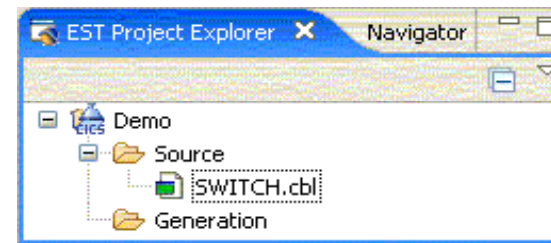
- On Import source files panel click Remote
- Select remote file from data set list



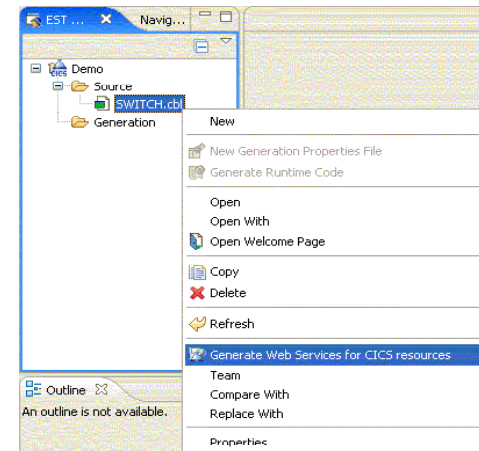
- Click Finish to import file



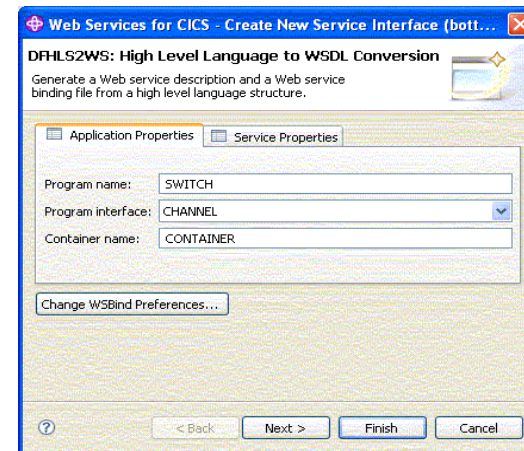
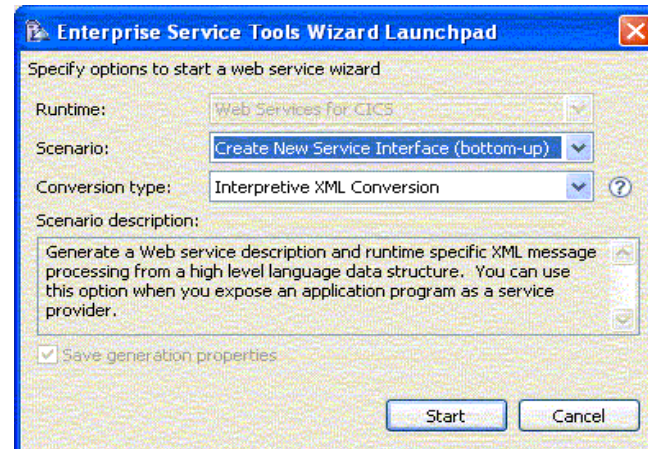
- File has been imported



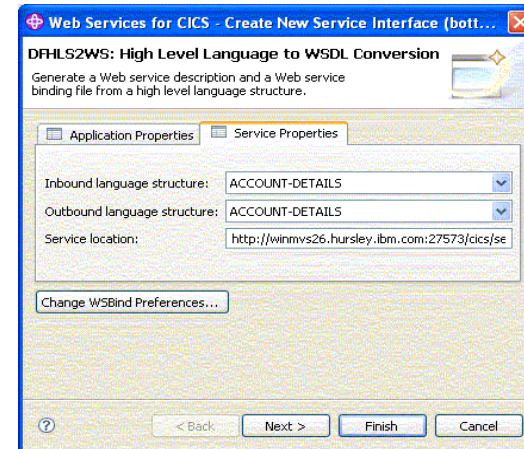
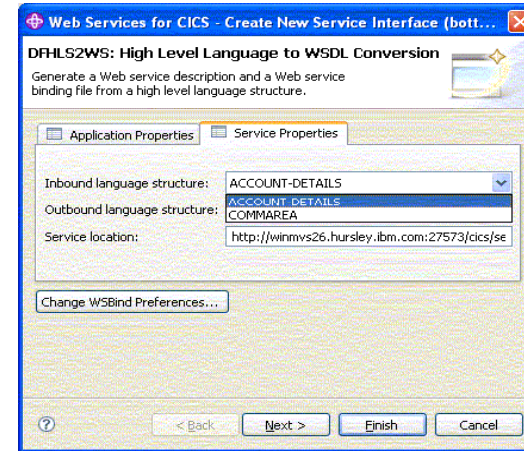
- Right click and select Generate Web Services for CICS resources



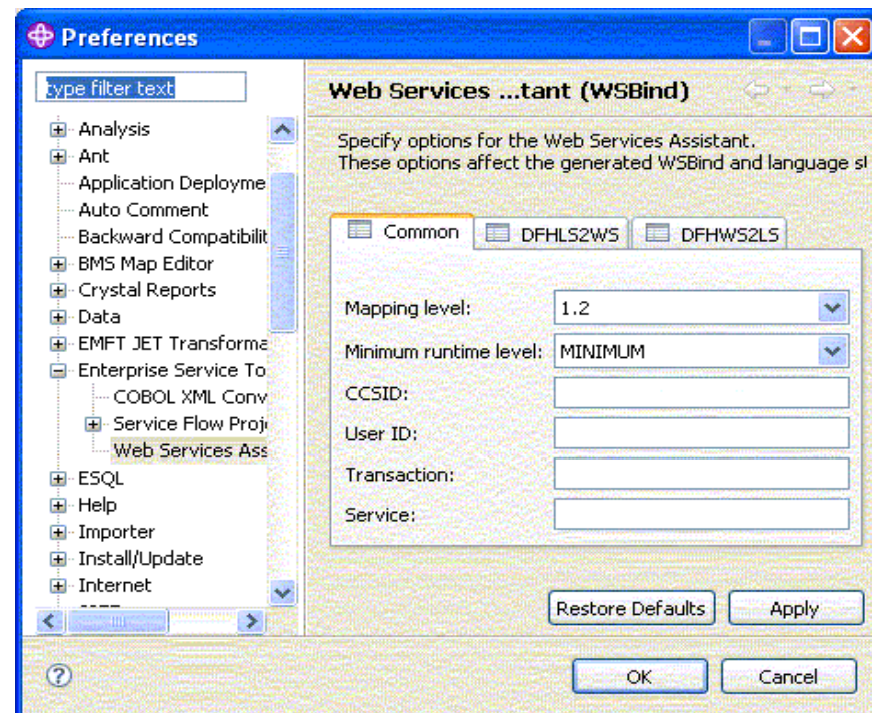
- Select Interpretive XML Conversion and press Start button
- Choose CHANNEL as program interface type and input a CONTAINER name



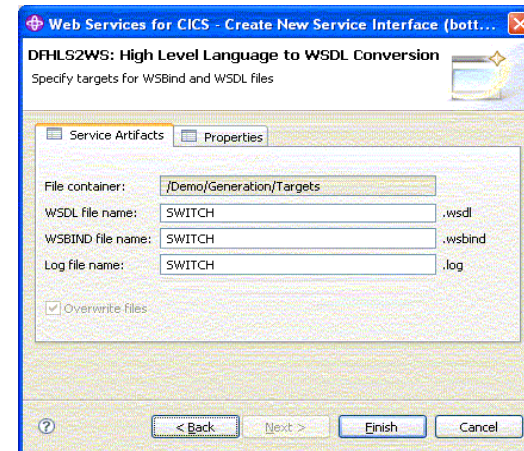
- On Service Properties tab select appropriate input and output language structures from source program
- Input IP address and port of TCPIP service in CICS region



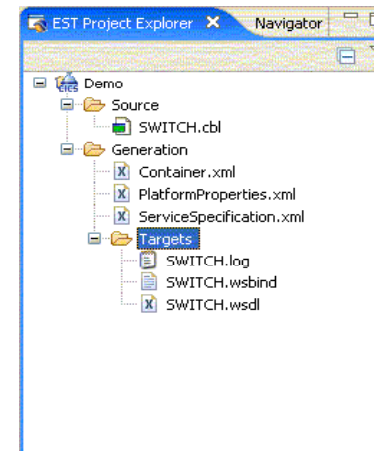
- Clicking on Change wsbind Preferences allows selection of wsbind file properties



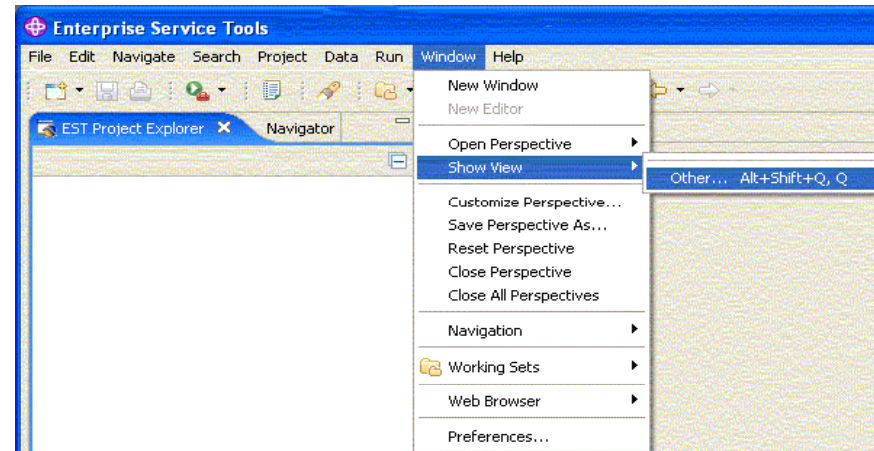
- Click Next to input generated artefact names



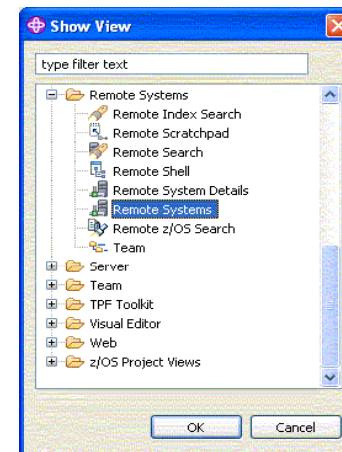
- New artefacts appear in EST Project Explorer window



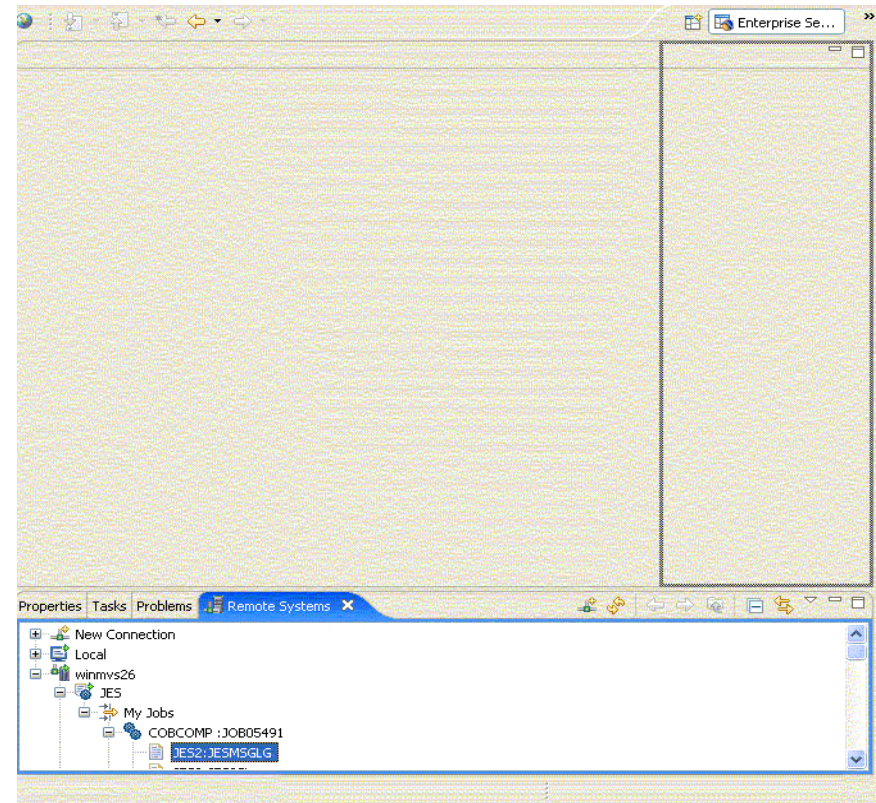
- Click on Show View dropdown



- Select Remote Systems from list



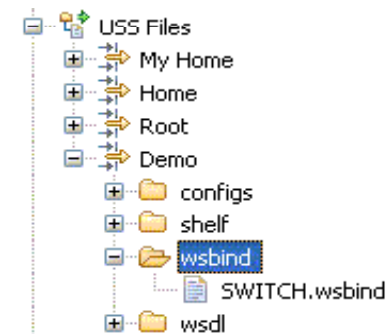
- Move Remote Systems window to make better use of display



- Drag SWITCH.wsbind from EST Project Explorer window to Remote Systems window

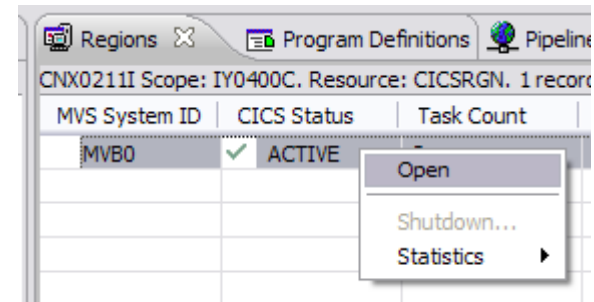


- wsbind file has been copied to USS



Manage using the CICS Explorer

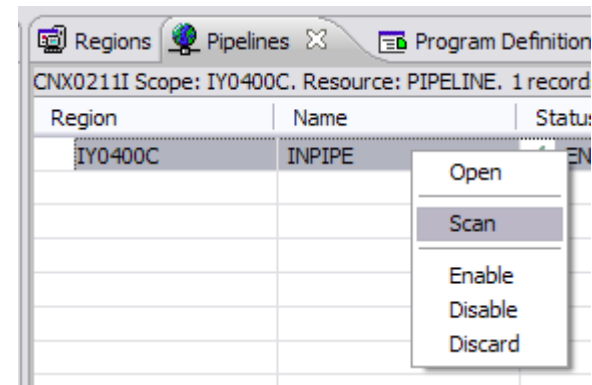
- Select appropriate CICS region



- Change DEBUG status to NODEBUG



- Select Pipeline tab and perform pipeline scan to deploy new Web service in CICS



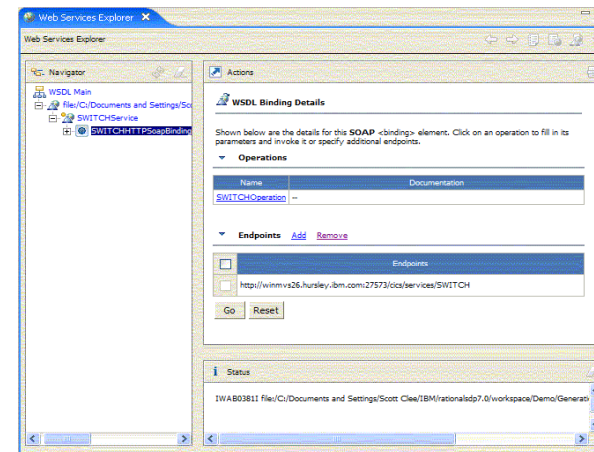
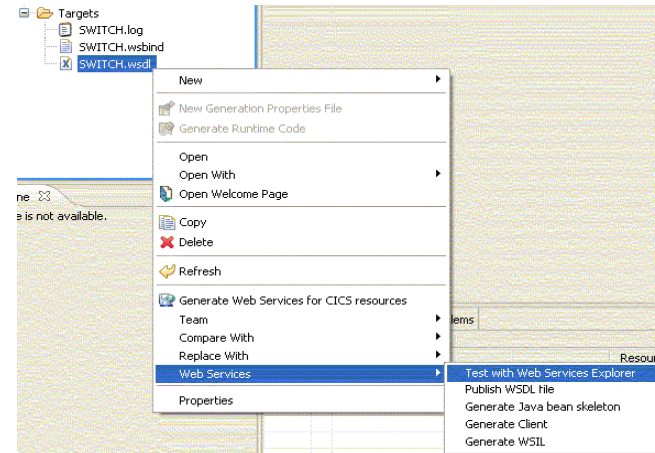
- Select Web Services tab see your deployed Web service

The screenshot shows the 'Web Services' tab in the IBM CICS console. The table below shows the web service details:

Region	Name	State	Use Count
IY0400C	SWITCH	✓ INSERVICE	0

Test the Web service

- In Enterprise Service Tools perspective right click on generated WSDL and select Test with Web Services Explorer
- Select SWITCHOperation from list



- Input data for SOAP request and click Go

Invoke a WSDL Operation [Source](#)

Enter the parameters of this WSDL operation and click Go to invoke.

Endpoints
http://winmvs26.hursley.ibm.com:27573/cics/services/SWITCH

SWITCHOperation

account_details

account_number unsignedInt
12345678

account_type string
A

balance int
0

Go Reset

- Observe response data

Status [Source](#)

SWITCHOperationResponse

account_details

account_number (unsignedInt): 12345678

account_type (string):

balance (int): 12345

- Click on Source to view request and response XML

```
<?xml version="1.0" encoding="UTF-8" ?>
- <soapenv:Envelope
  xmlns:q0="http://www.SWITCH.0.Request.com"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/en"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema
  -instance">
- <soapenv:Body>
- <q0:SWITCHOperation>
  - <q0:account_details>

    <q0:account_number>12345678</q0:account_n

    <q0:account_type>A</q0:account_type>
    <q0:balance>0</q0:balance>
  </q0:account_details>
</q0:SWITCHOperation>
</soapenv:Body>
</soapenv:Envelope>
```

```
- <SOAP-ENV:Envelope xmlns:SOAP-
  ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:q0="http://www.SWITCH.0.Request.com"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/en"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema
  -instance">
- <SOAP-ENV:Body>
- <SWITCHOperationResponse
  xmlns="http://www.SWITCH.0.Response.com">
  - <account_details>

    <account_number>12345678</account_number>
    <account_type />
    <balance>12345</balance>
  </account_details>
</SWITCHOperationResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

RDz screen shots

z/OS Projects perspective

The screenshot displays the z/OS Projects IDE interface. The main editor window shows the source code for TESTPROG.cbl, which includes sections for WORKING-STORAGE, LINKAGE, and PROCEDURE DIVISION. The PROCEDURE DIVISION contains logic for setting account details and linking to a SWITCH program.

```

Line 20      Column 1      Insert
-----+*A-1-B-----2-----3-----4-----5-----6-----+
WORKING-STORAGE SECTION.
*
01  ACCOUNT-DETAILS.
   03  ACCOUNT-NUMBER   PIC 9(8) .
   03  ACCOUNT-TYPE    PIC X(1) .
   03  BALANCE          PIC S9(8) BINARY.
*
LINKAGE SECTION.
*
PROCEDURE DIVISION.
*
TESTPROG-MAIN SECTION.
*
TESTPROG-00.
*
* Set account number and account type
*
MOVE '12345678' TO ACCOUNT-NUMBER.
MOVE 'A' TO ACCOUNT-TYPE.
MOVE 0 TO BALANCE.
*
EXEC CICS PUT CONTAINER('CONTAINER')
      CHANNEL('CHANNEL')
      FROM(ACCOUNT-DETAILS)
END-EXEC.
*
EXEC CICS LINK PROGRAM('SWITCH')
    
```

The left pane shows the project structure for 'Demo' under 'winmvs26'. The bottom-left pane shows the program outline for TESTPROG. The bottom-right pane shows the 'Remote System Details' table:

Name	Parent profile	Remote system ...	Connection status	Host name	Default User ID	Description
Local	Hursley	Local	Some subsystem...	LOCALHOST	laptop (Inherited)	
winmvs26.hursley.ibm...	Hursley	z/OS	Some subsystem...	WINMVS26.HUR...	scottc	

Debug perspective

The screenshot displays the IBM CICS debug perspective for the program SCOTT.C.SYSDEBUG(SWITCH). The interface includes a menu bar, a toolbar, and several panes:

- Debug Console:** Shows the current thread (Thread: 1 (Runnable)) and process (Process: 514797880 Program: SWITCH).
- Variables:** A table showing the current state of variables:

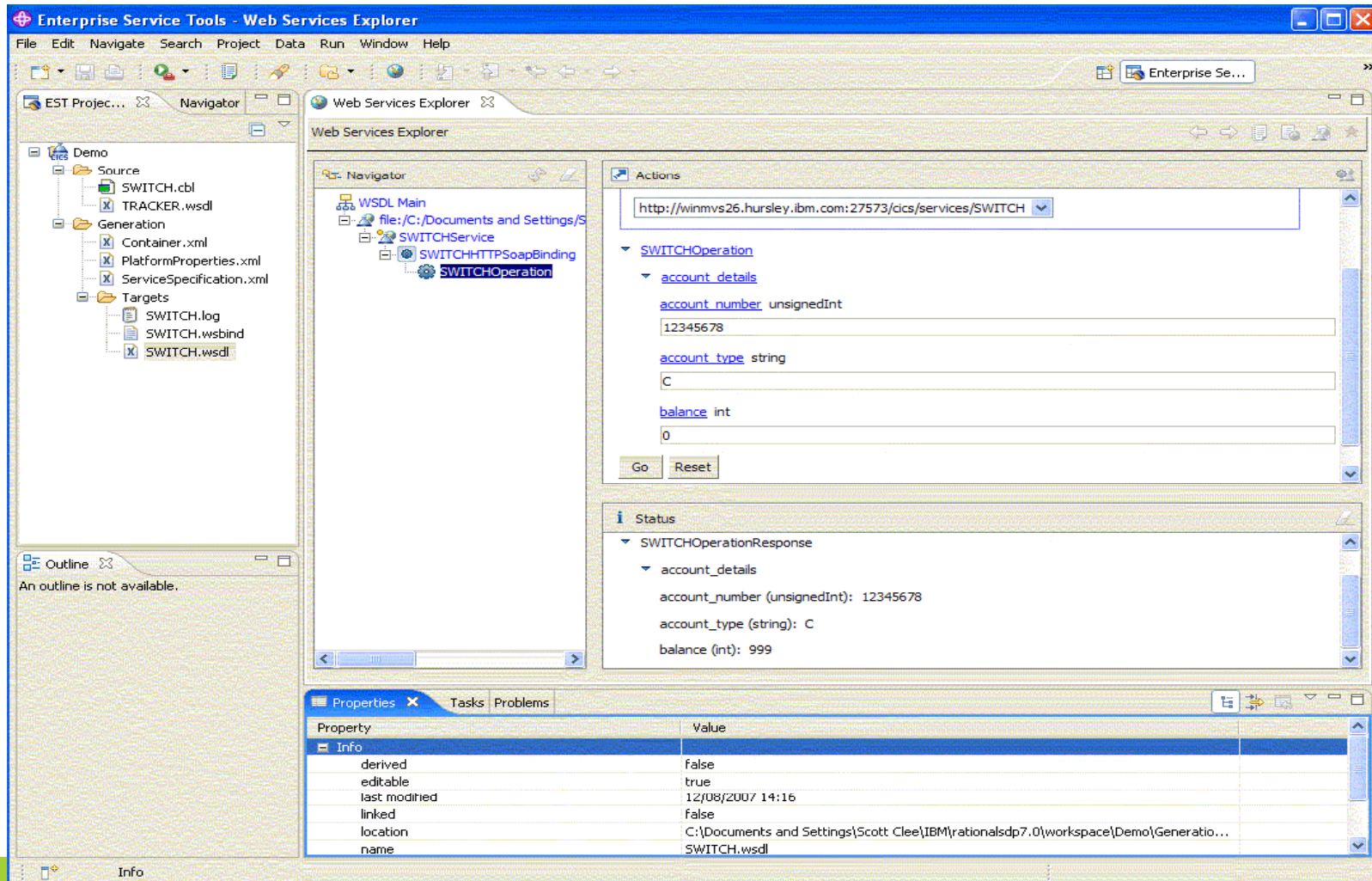
Name	Value
ACCOUNT-DETAILS	
ACCOUNT-NUMBER	12345678
ACCOUNT-TYPE	'C'
BALANCE	+0000000000
COMMAREA	
REQUEST	
- Source Code:** Shows the following code snippet:


```

52 * Link to different bank programs according to account type
53 *
54 IF ACCOUNT-TYPE EQUAL 'A'
55   PERFORM LINK-BANKA
56 ELSE IF ACCOUNT-TYPE EQUAL 'B'
57   PERFORM LINK-BANKB
58 ELSE IF ACCOUNT-TYPE EQUAL 'C'
59   PERFORM LINK-BANKC
60 END-IF.
61 *
62 * Put the structure in the response container
63 *
64 EXEC CICS PUT CONTAINER('CONTAINER')
65   FROM(&ACCOUNT-DETAILS)
            
```
- Breakpoints:** A breakpoint is set on the statement [SCOTT.C.SYSDEBUG(SWITCH)].
- Monitors:** A memory monitor for ACCOUNT-DETAILS is active, showing a hex dump:

Address	0 - 3	4 - 7	8 - B	C - F
1EAFBCC0	00000000	00000000	F1F2F3F4	F5F6F7F8
1EAFBCD0	C3000000	00000000	00000000	00000000
1EAFBCE0	00000000	00000000	00000000	00000000
1EAFBCF0	00000000	00000000	00000000	00000000
1EAFBD00	00000000	00000000	00000000	00000000
1EAFBD10	00000000	00000000	00000000	00000000

Enterprise Service Tools perspective



Debugging a CICS Java program

The screenshot displays an IDE interface for debugging a Java program. The main components are:

- Debug Console:** Shows the execution stack for a thread named "Thread [HELOWORL.TASK69.HWRL] (Suspended (breakpoint at line 14 in HelloWorld))". The current line of execution is "HelloWorld.main(CommAreaHolder) line: 14".
- Variables Window:** Displays the state of local variables:

Name	Value
commarea	CommAreaHolder (id=22)
gas	null
value	byte[0] (id=26)
- Code Editor:** Shows the source code of "HelloWorld.java". The current line of execution is highlighted:


```

import com.ibm.cics.server.Channel;

public class HelloWorld
{
    public static void main(CommAreaHolder commarea)
    {
        try
        {
            final Channel channel = Task.getTask().createChannel("MYCHANNEL");
            final Container container = channel.createContainer("DATA");

            container.put("HelloWorld");
        }
        catch (Exception e) {}
    }
}
            
```
- Outline Window:** Shows the project structure, including "com.ibm.test", "import declarations", and "HelloWorld 1.1 (ASCII -kkv)".

Benefits of RDz over ISPF programming

- Doesn't tie up your TSO userid
- Simultaneously edit multiple programs in varying languages
- Windowing style copy and paste
- Auto-complete of EXEC CICS commands
- Drag and drop files between PC, MVS & USS
- More screen real estate
- Code, debug, Web service enable, create service flows (SFF) all from the one tool
- The whole of this demo was being done remotely from a machine in Hursley!

Summary

- Introduce Rational Developer for System z (RDz)
- Configuring Debug Tool for CICS
- Live Demo!
 - z/OS Projects perspective
 - Debug perspective
 - Enterprise Service Tools perspective
 - CICS Explorer
 - CADP
- Benefits of RDz over ISPF programming

References

- CICS TS 4.1 InfoCenter
 - <http://publib.boulder.ibm.com/infocenter/cicsts/v4r1>
- Rational Developer for System z
 - <http://www-306.ibm.com/software/awdtools/rdz/>
- Debug Tool for System z
 - <http://www-306.ibm.com/software/awdtools/debugtool/>
- Problem determination tools
 - <http://www-306.ibm.com/software/awdtools/deployment>
- CICS Explorer
 - <http://www-306.ibm.com/software/htp/cics/explorer/>