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 taking 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.

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 EQACCSD in Debug Tool's SEQASAMP data set
- Follow instructions in EQACCSD 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),`

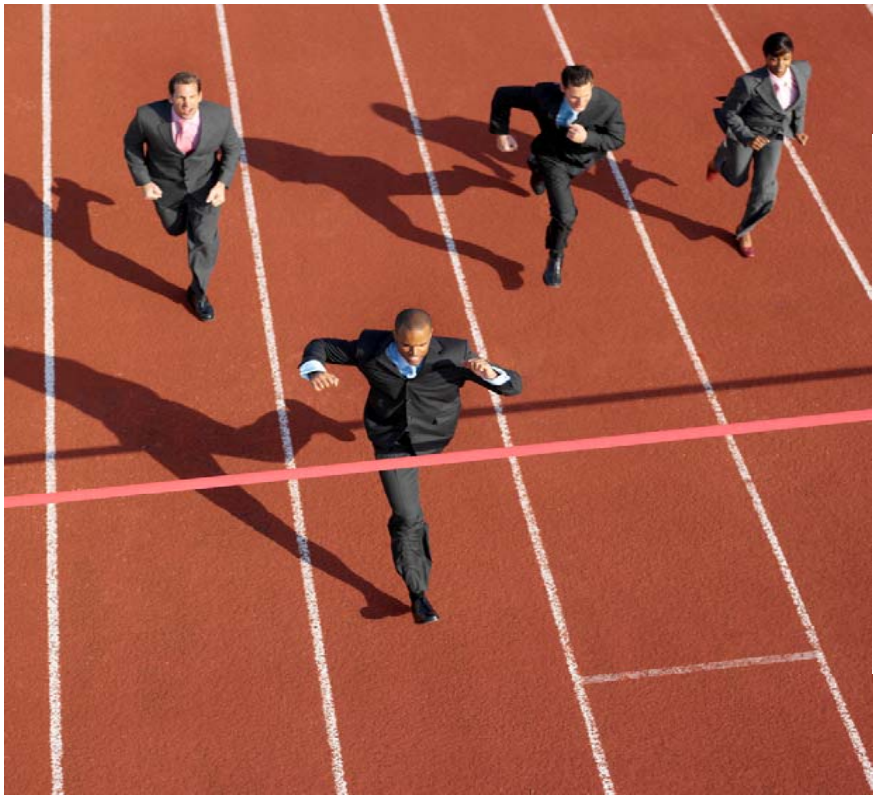
Enable debug mode in CICS

- 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

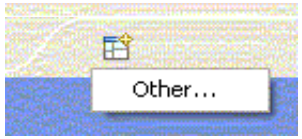
Live demo!



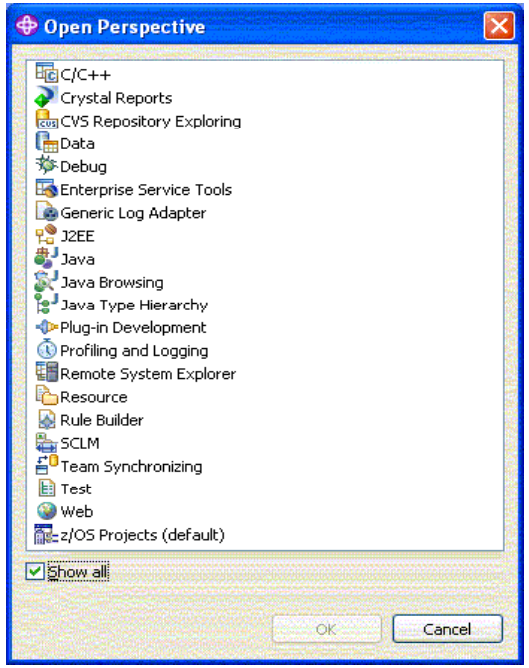
- The following pages are for reference purposes

Perspectives

- Click on open perspective button

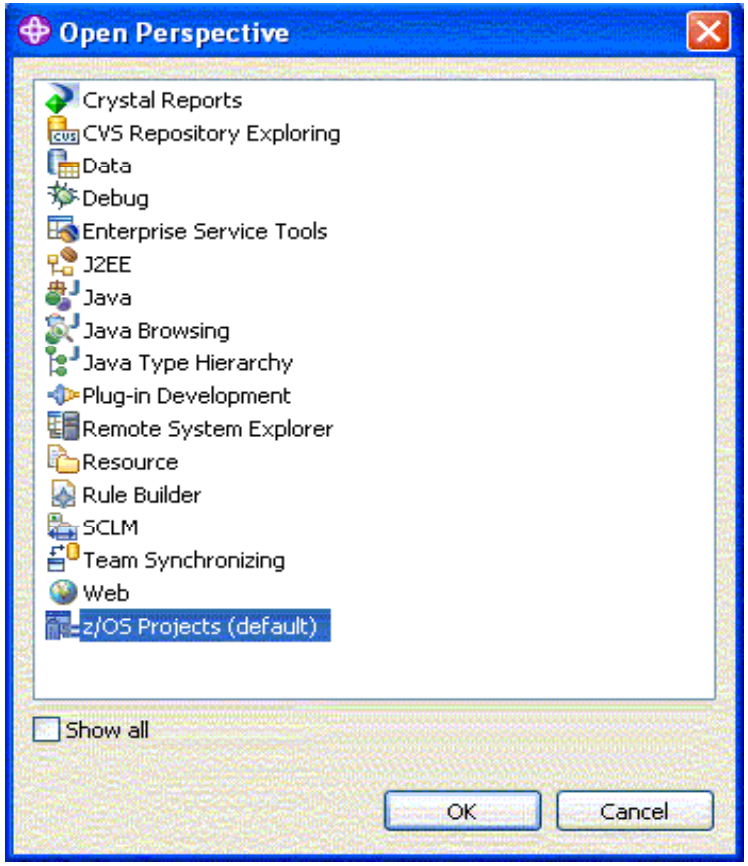


- Select a perspective from the list

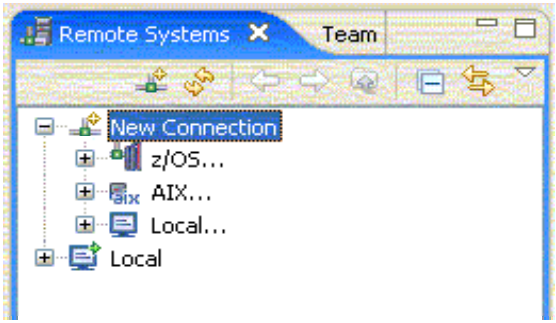


z/OS Projects

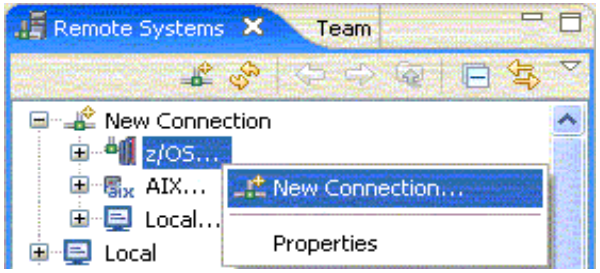
- Select the z/OS Projects perspective



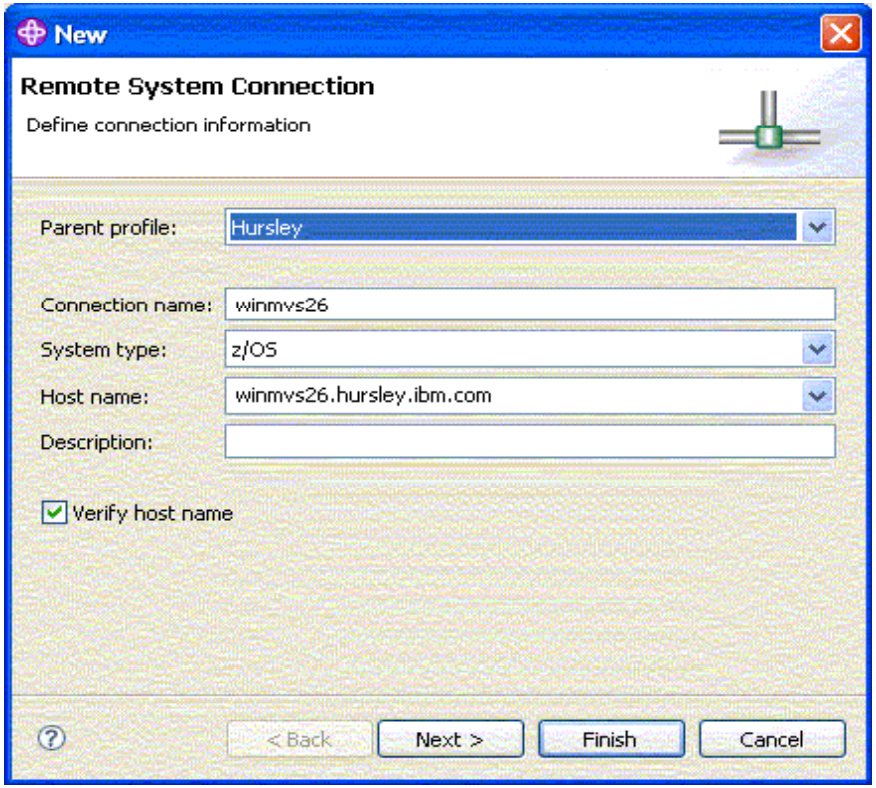
- Remote systems panel opens up



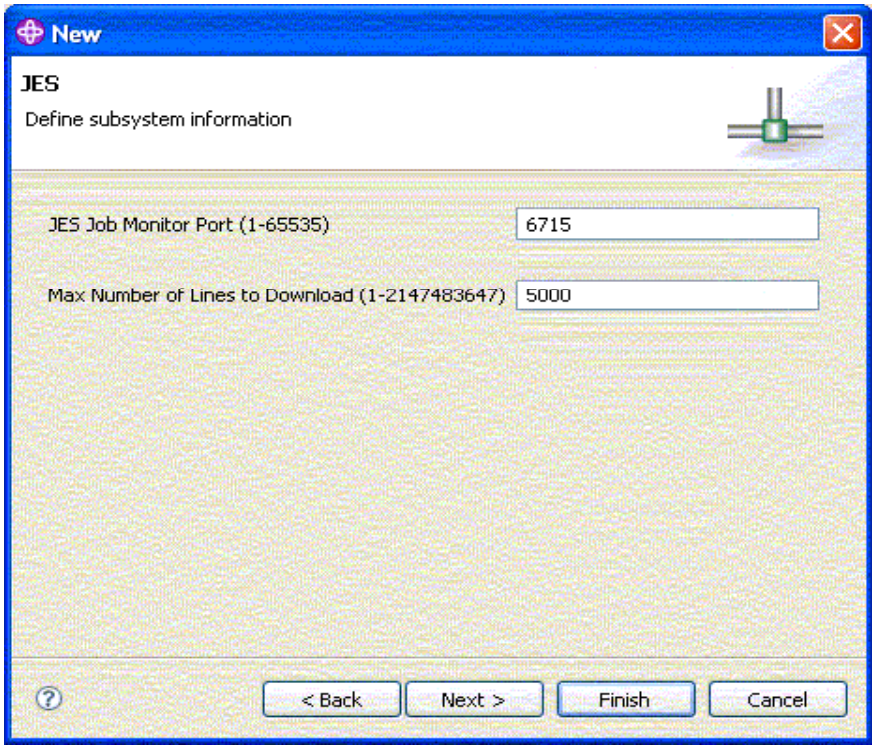
- Right click on z/OS and create a new connection



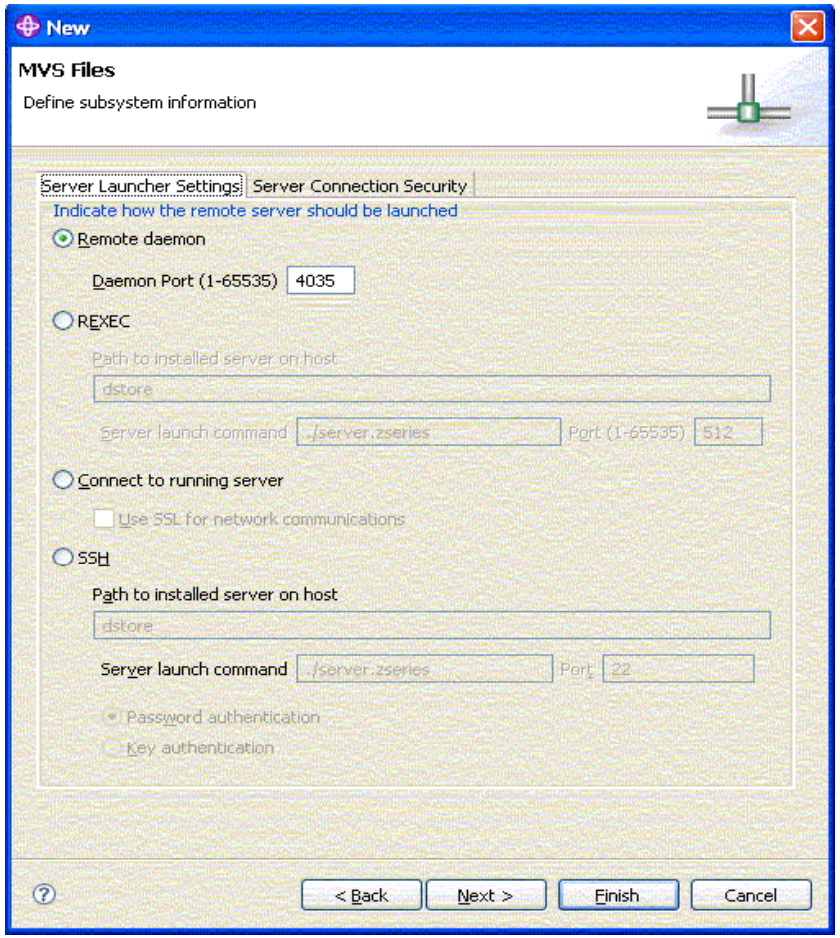
- Enter connection details and press next



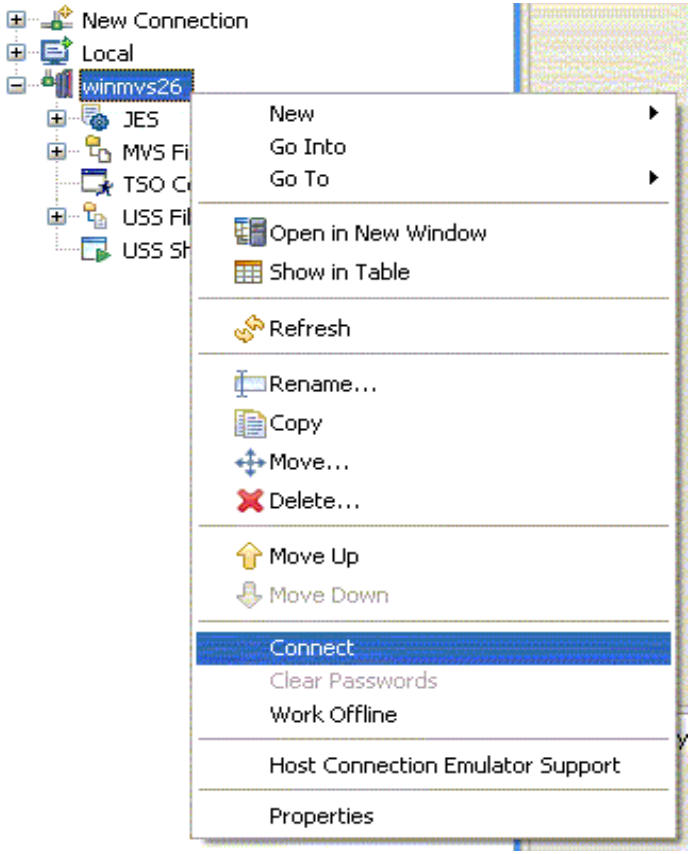
- Enter port numbers



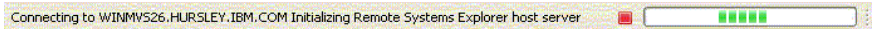
- Select connection type



- Right click on system icon and connect

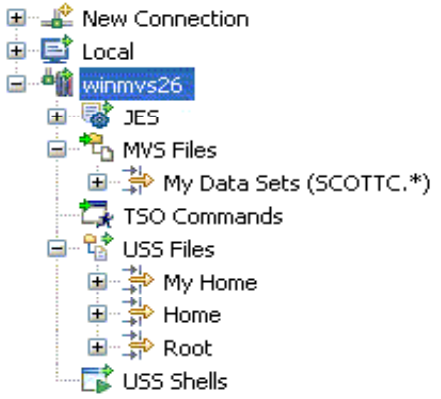


- Input username and password for system (TSO logon credentials)

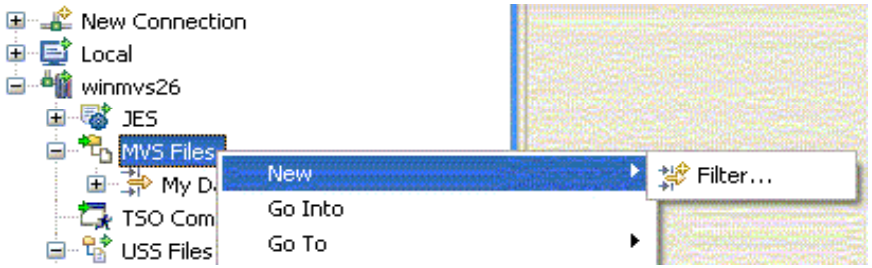


- Observe connecting status

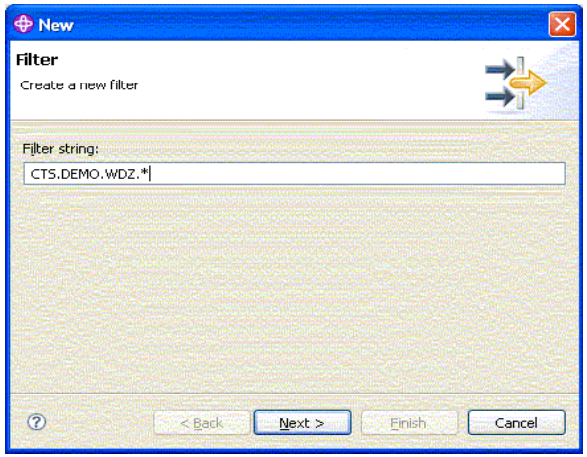
- Once connected expand the MVS Files and USS Files sections



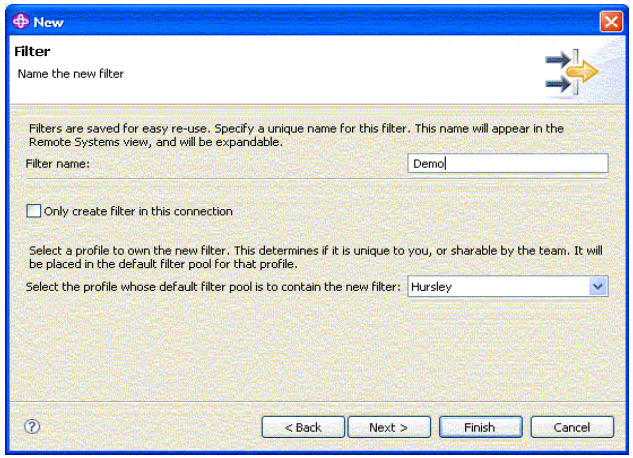
- Right click on MVS Files and create new filter



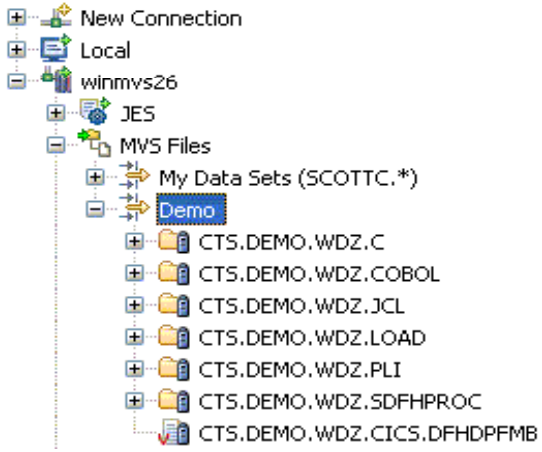
- Input filter criteria



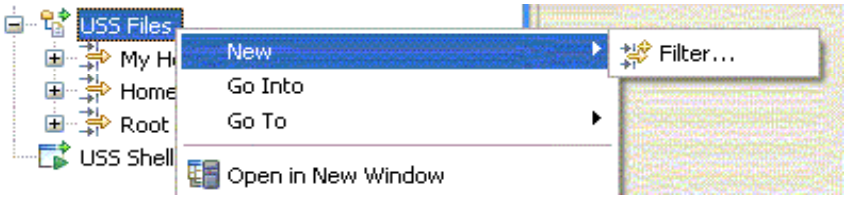
- Specify a name for the filter



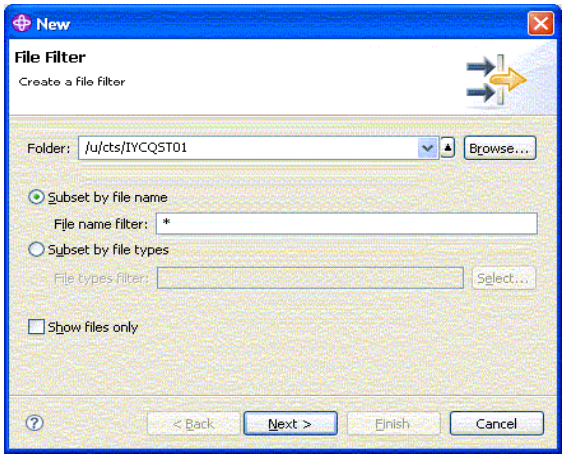
- Expand the filter icon to see data sets



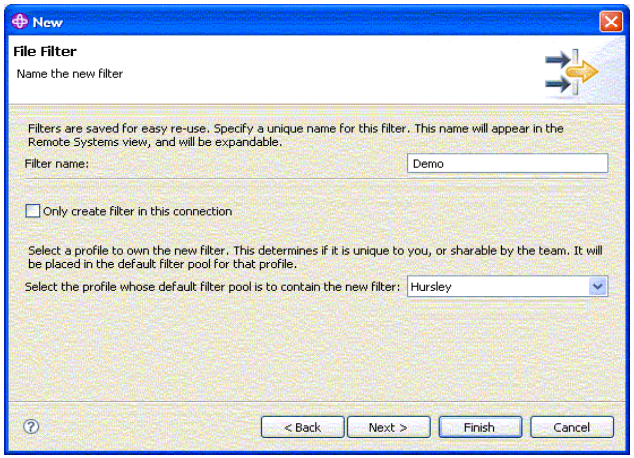
- Right click on USS Files and create new filter



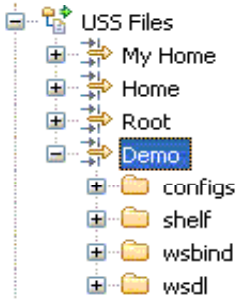
- Input filter criteria



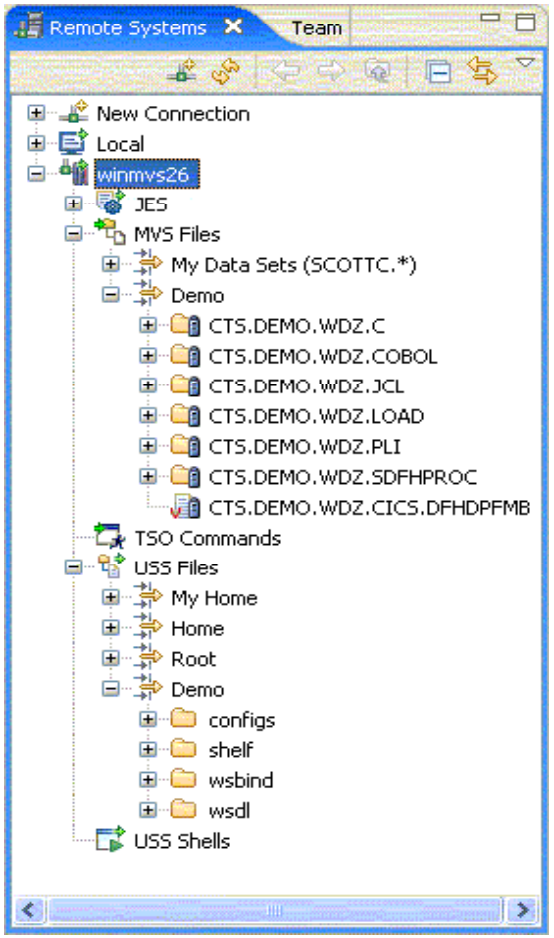
- Specify a name for the filter



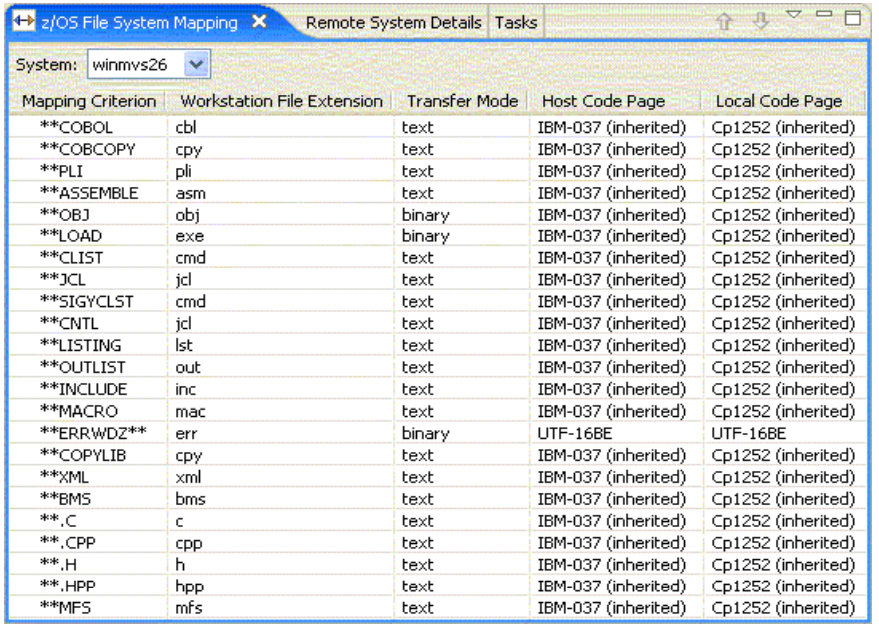
- Expand the filter icon to see data sets



- Now have connection with filters in place



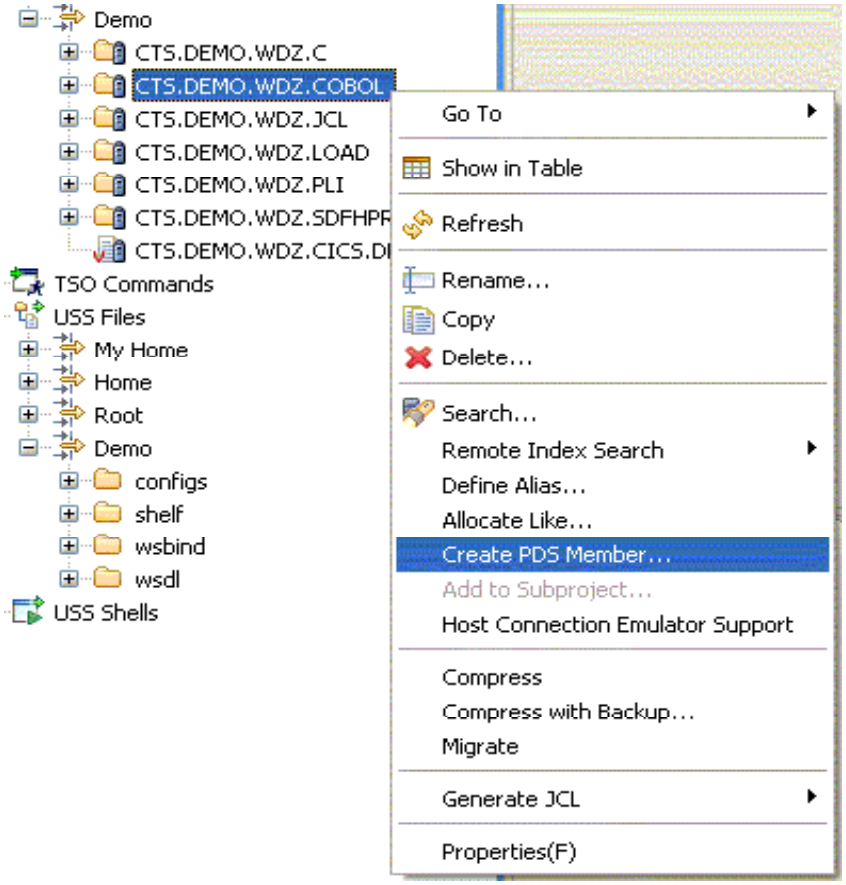
- Observe file system mappings window
  - Includes code page information



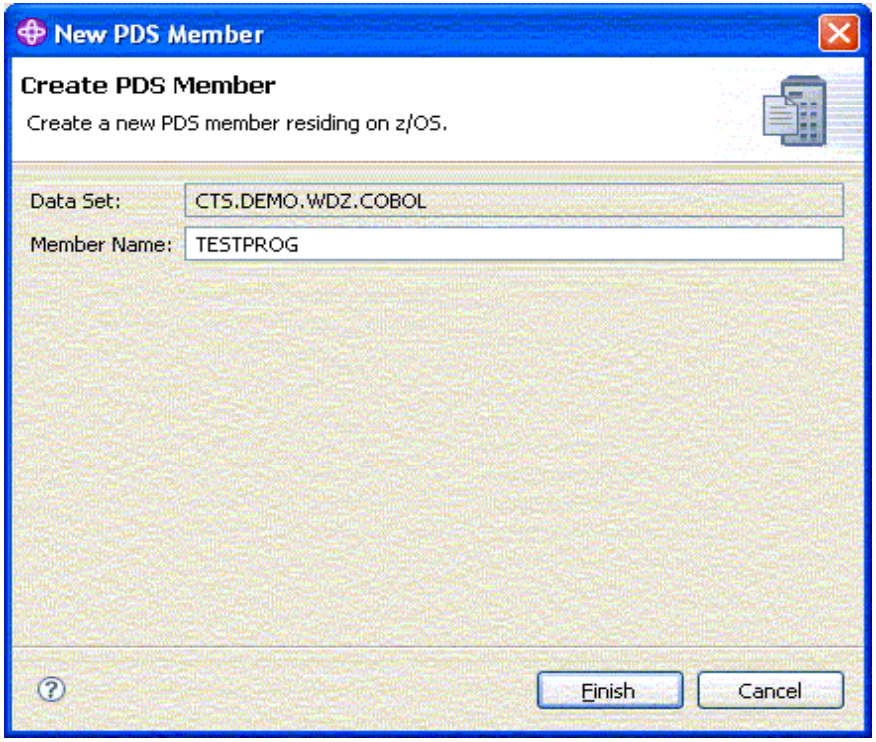
Mapping Criterion	Workstation File Extension	Transfer Mode	Host Code Page	Local Code Page
**COBOL	cbl	text	IBM-037 (inherited)	Cp1252 (inherited)
**COBCOPY	cpy	text	IBM-037 (inherited)	Cp1252 (inherited)
**PLI	pli	text	IBM-037 (inherited)	Cp1252 (inherited)
**ASSEMBLE	asm	text	IBM-037 (inherited)	Cp1252 (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

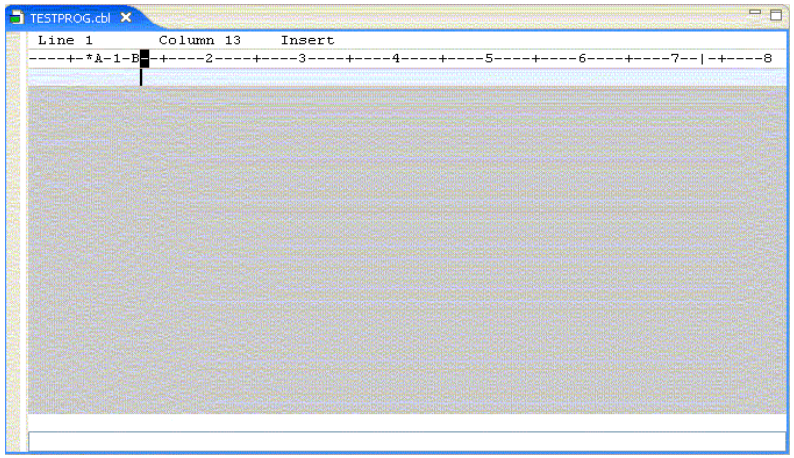
- Right click on data set and create a PDS member



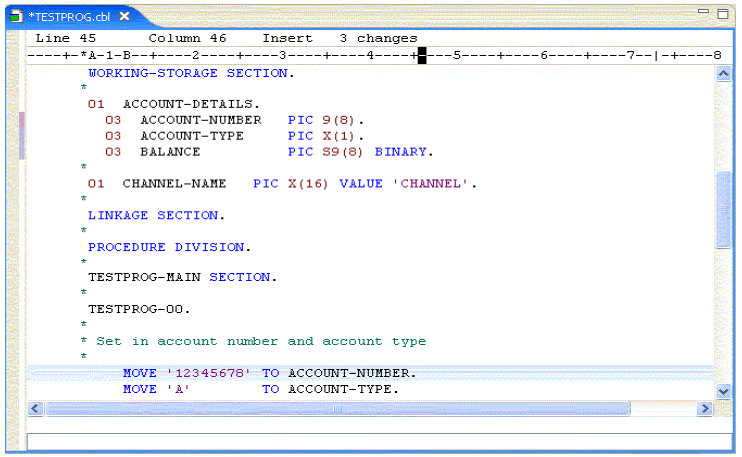
- Give the member a name



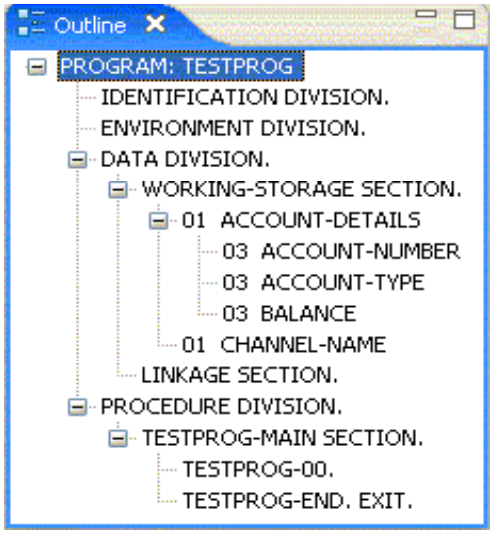
- Double click on TESTPROG to see empty file in editor



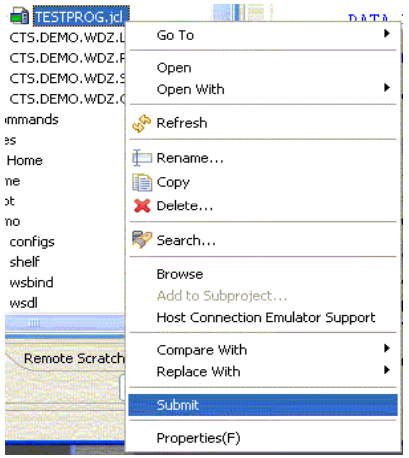
- Type in COBOL source code

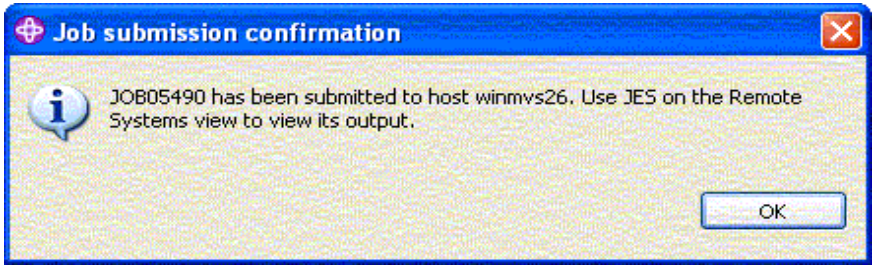


- Observe program window



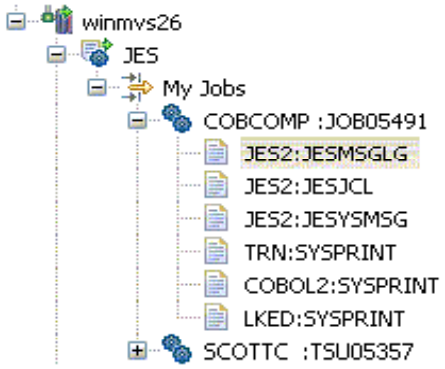
- Right click on compilation JCL and submit



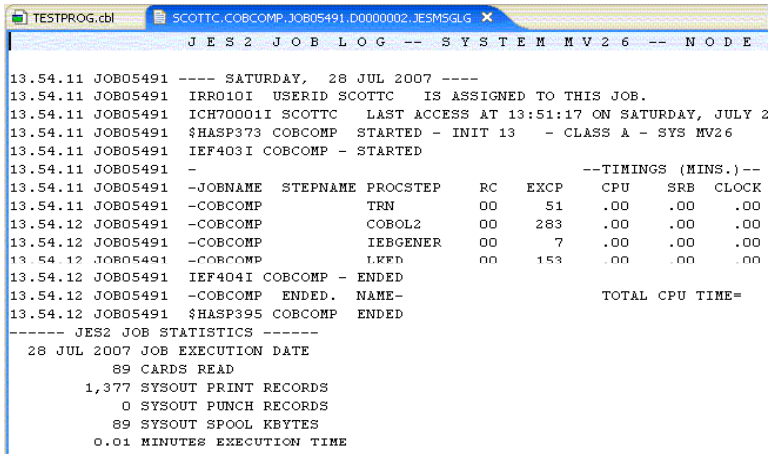


- Click OK on submission confirmation

- View job logs

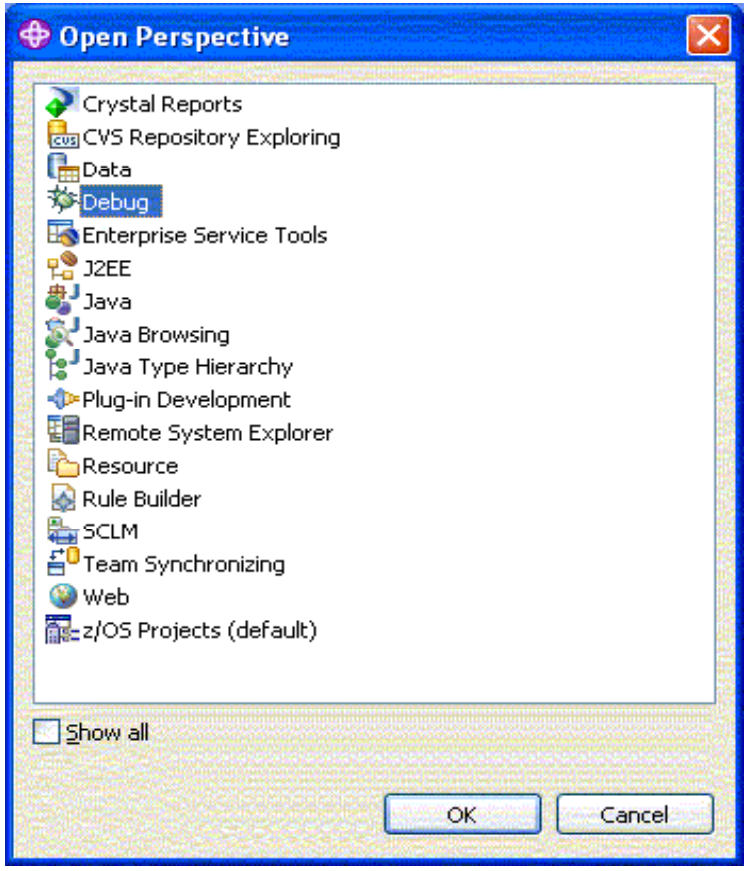


- Double click to see log information

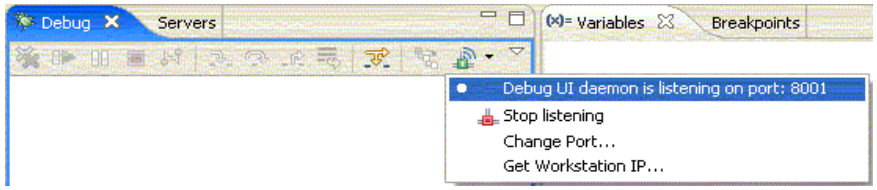


Debugging a COBOL program

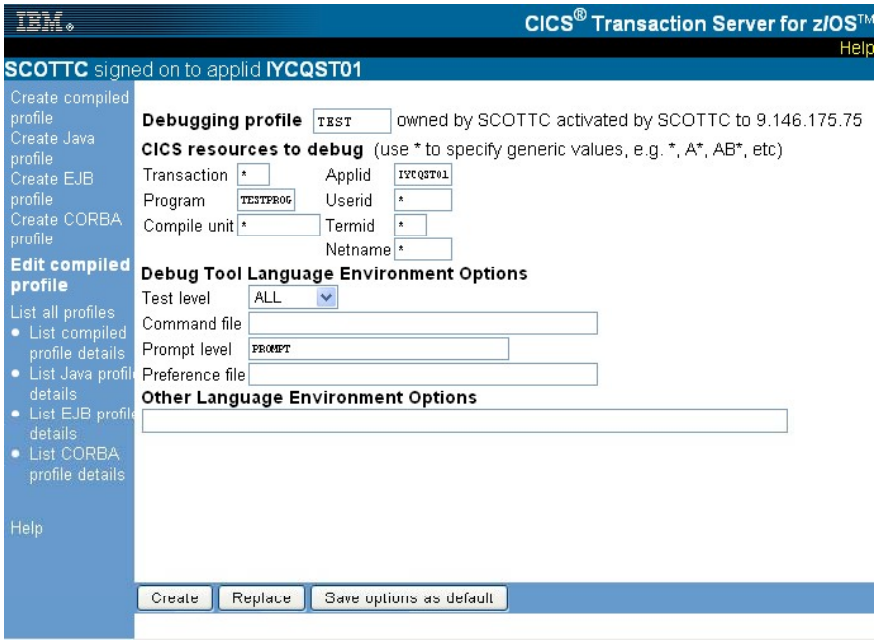
- Open up Debug perspective



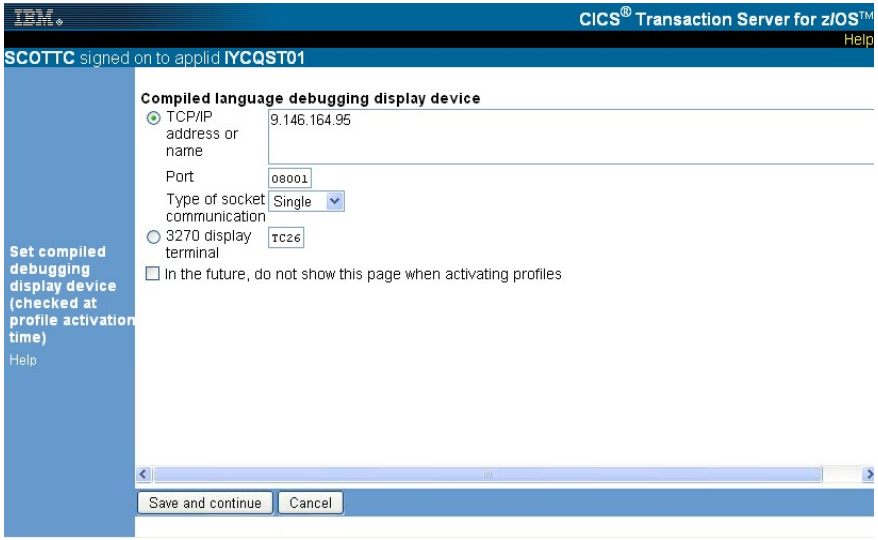
- Click on debug daemon icon and make note of the port it is listening on



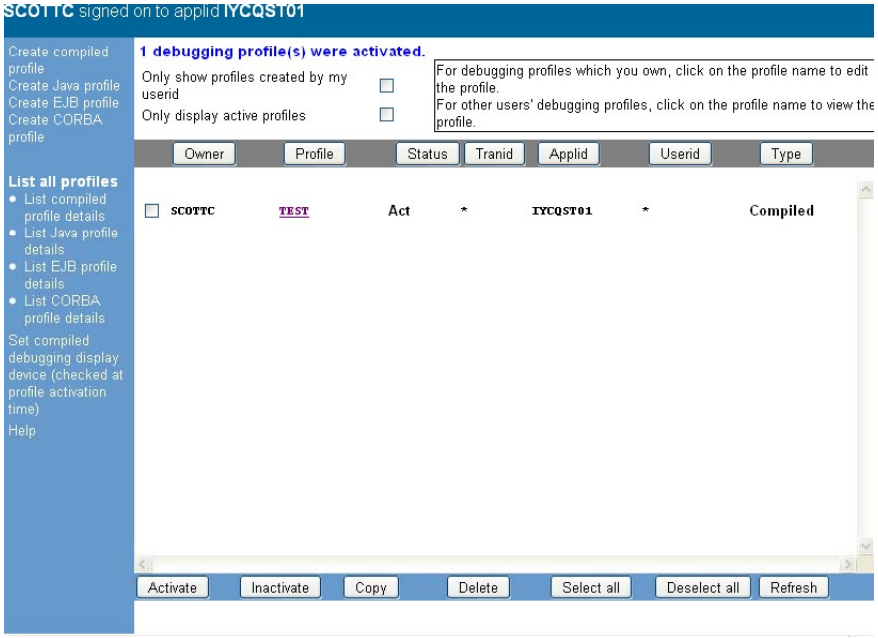
- Go to CADP Web interface and create a new debug profile



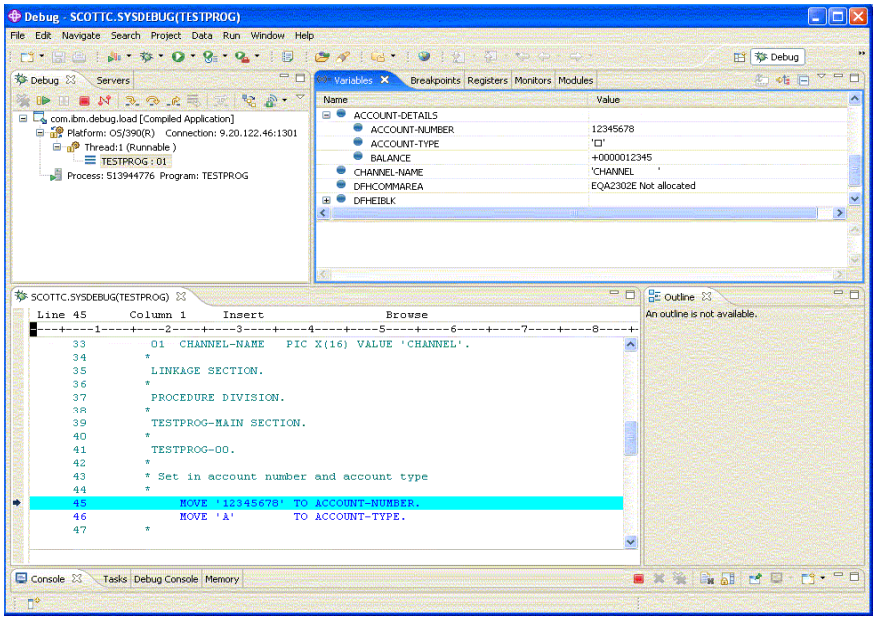
- Activate profile
  - Set session type to TCP and input IP address of machine running RDz
  - Match port number with that displayed on RDz debug daemon



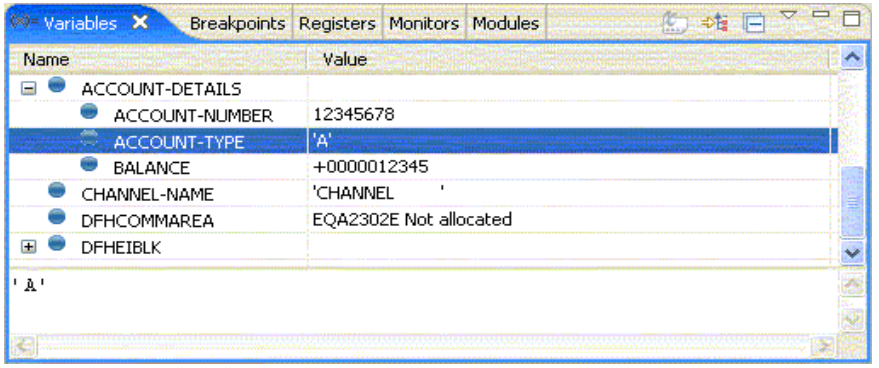
- Profile is active and ready to debug program



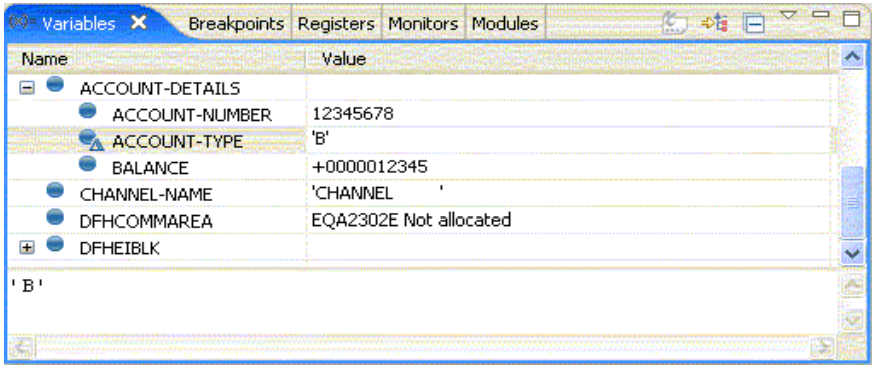
- As transaction starts it connects to RDz and debugger shows TESTPROG source code



- As variables are created their values appear in window



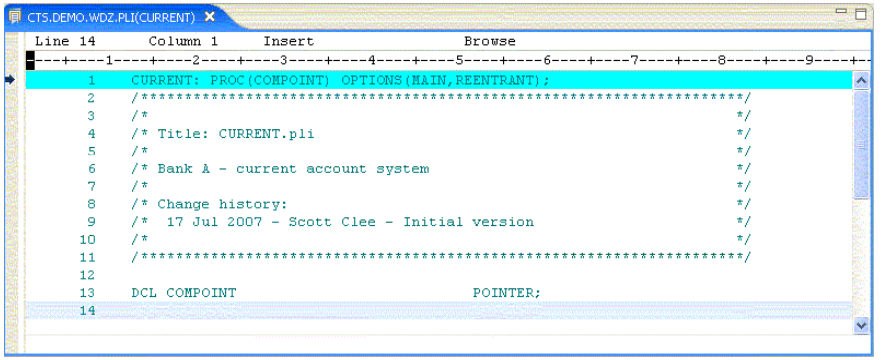
- Values can be modified in window and will be reflected in the running CICS program



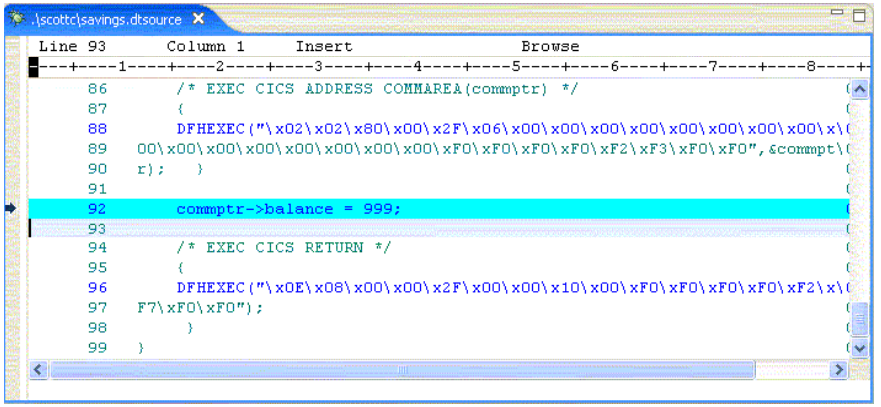
- Right click next to a line of source code to add a breakpoint

- Or choose Run To Location

- Example of CICS PL/I program being debugged

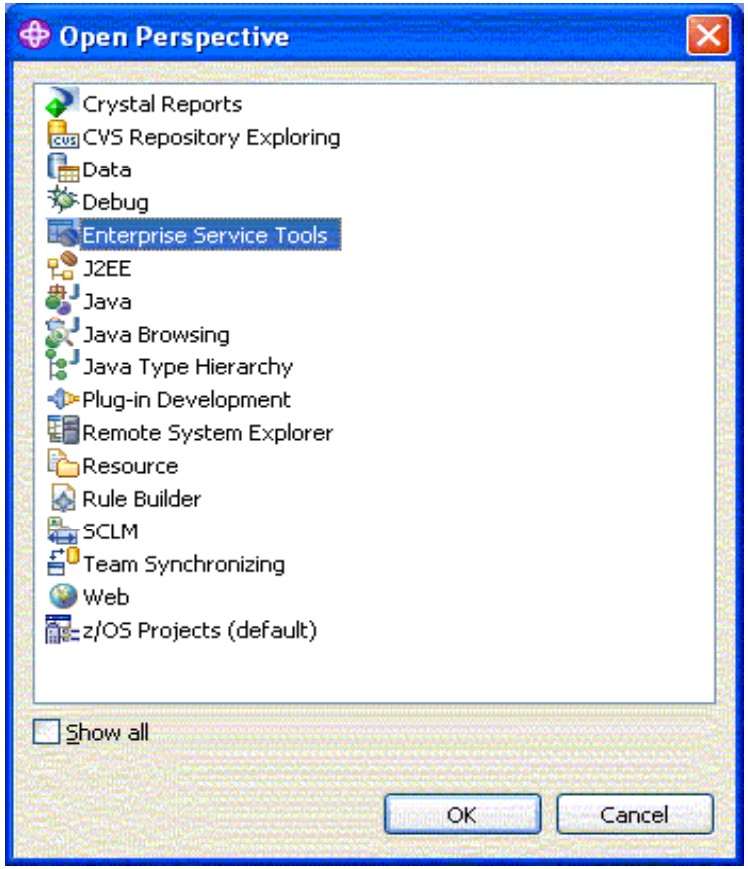


- Example of CICS C program being debugged

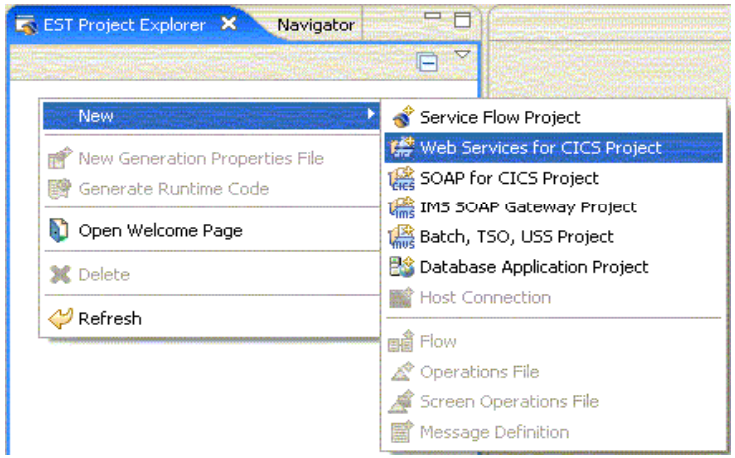


Expose as Web service

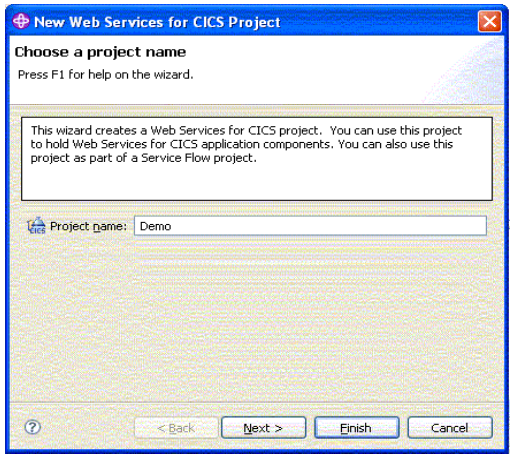
- Open Enterprise Service Tools perspective



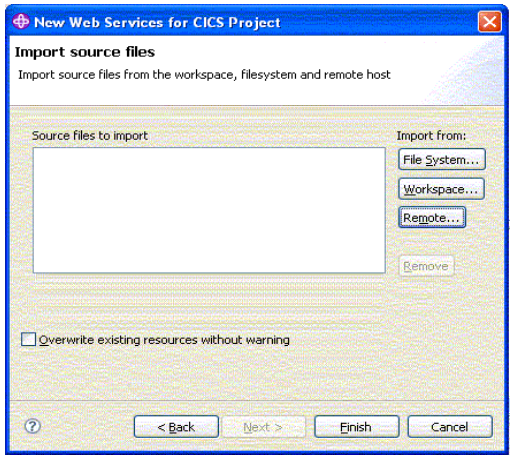
- Right click in EST Project Explorer window and create new Web Services for CICS Project



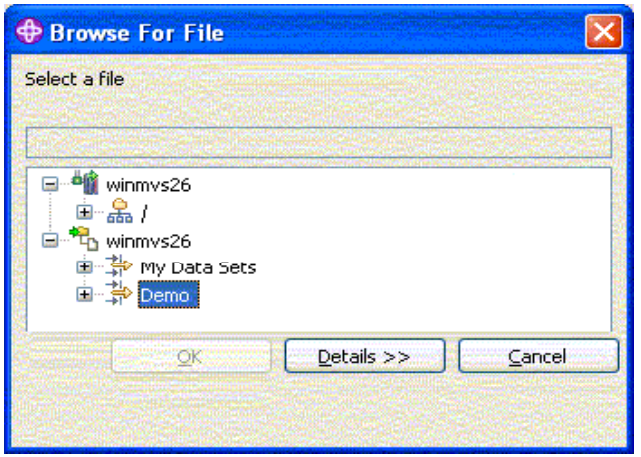
- Give the project a name



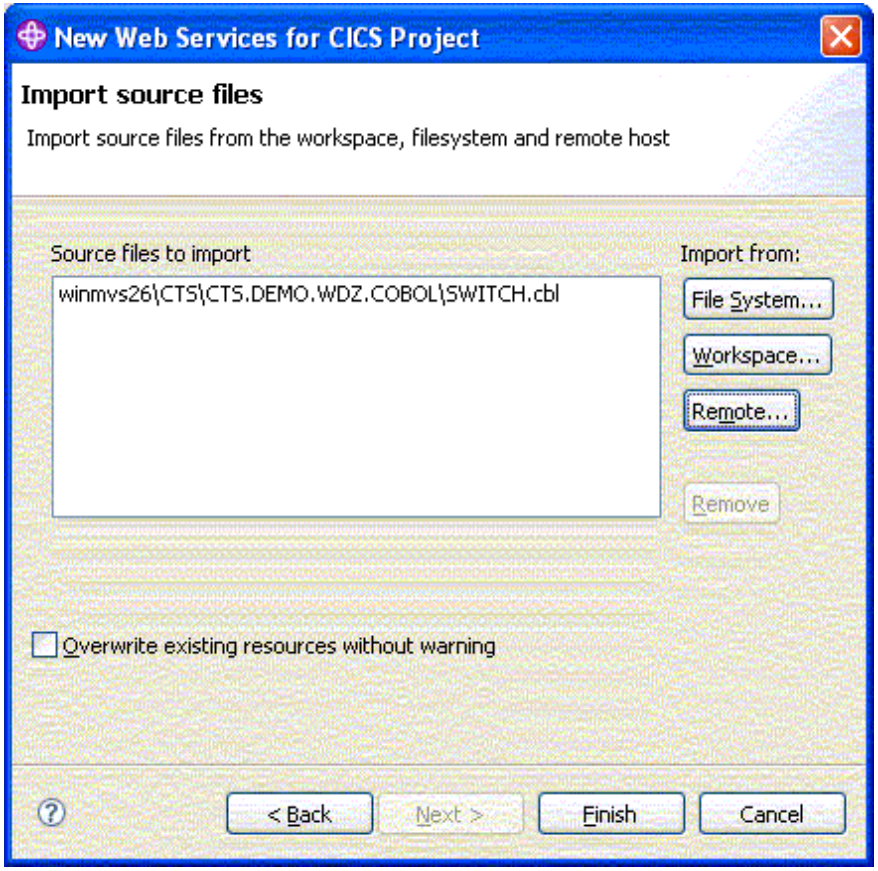
- On Import source files panel click Remote



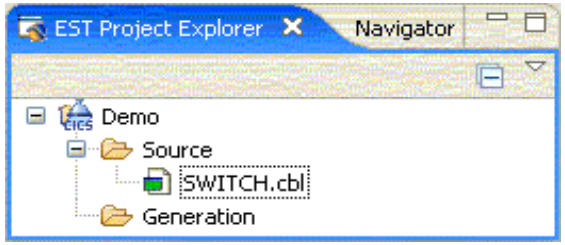
- Select remote file from data set list



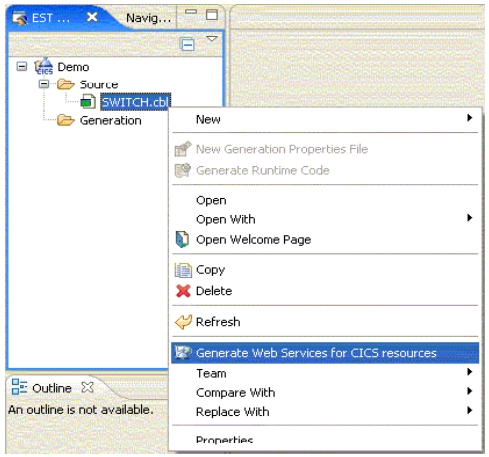
- Click Finish to import file



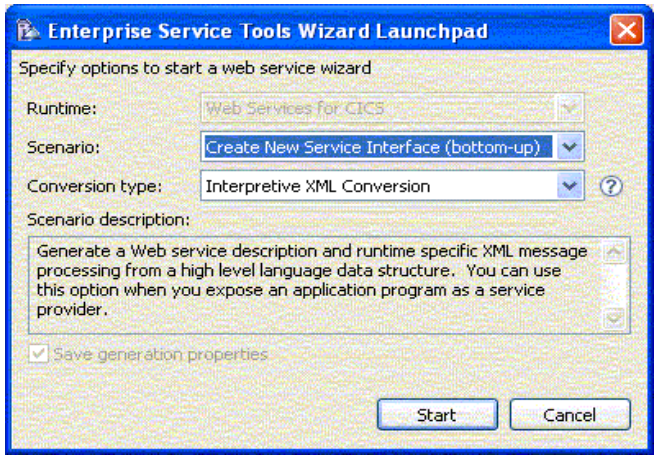
- File has been imported



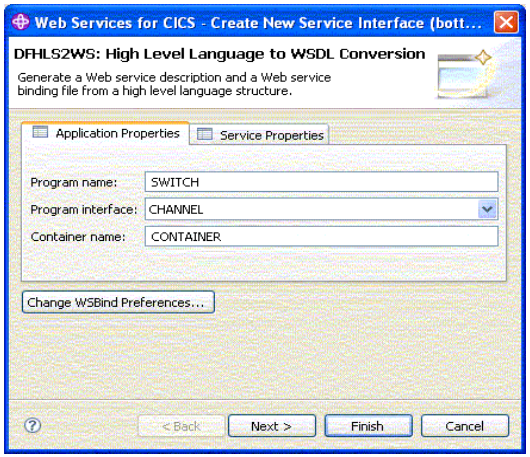
- Right click and select Generate Web Services for CICS resources



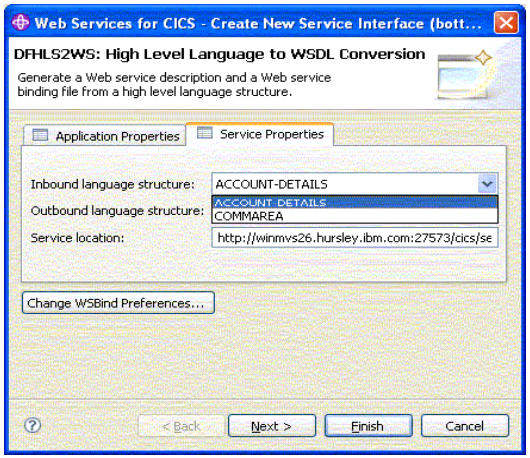
- Select Interpretive XML Conversion and press Start button



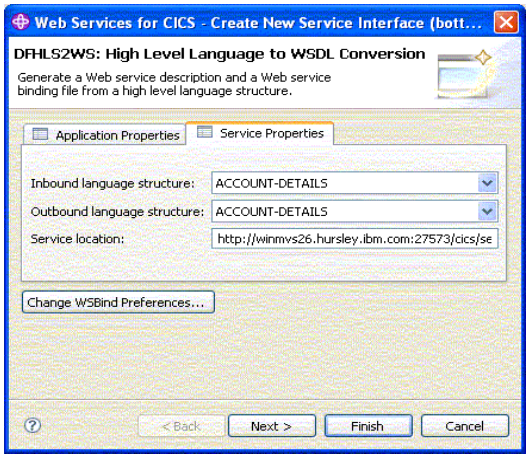
- Choose CHANNEL as program interface type and input a CONTAINER name



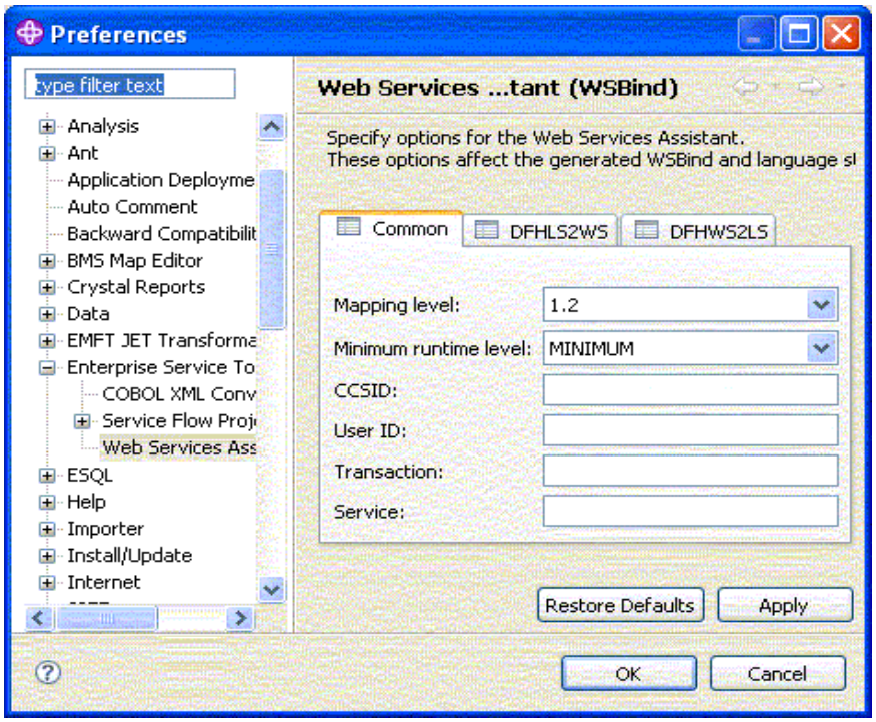
- On Service Properties tab select appropriate input and output language structures from source program



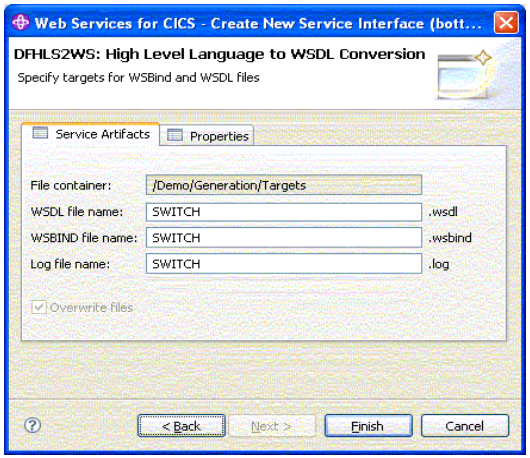
- Input IP address and port of TCPIP service in CICS region



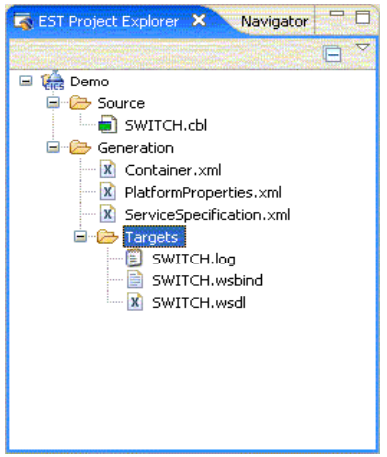
- Clicking on Change wsbind Preferences allows selection of wsbind file properties



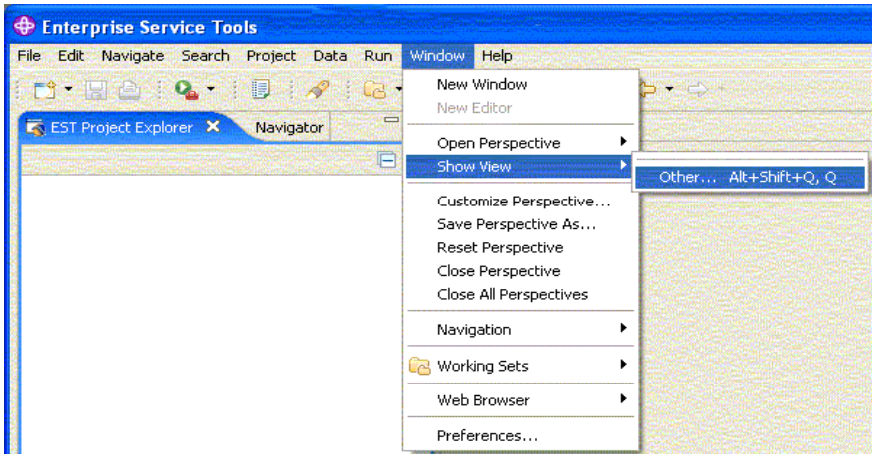
- Click Next to input generated artefact names



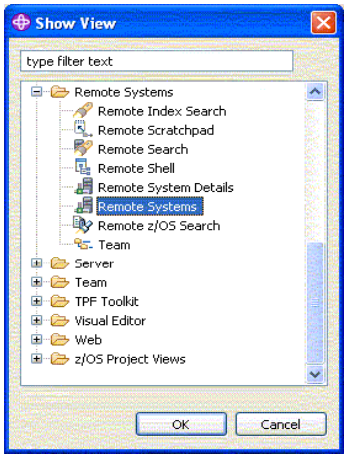
- New artefacts appear in EST Project Explorer window



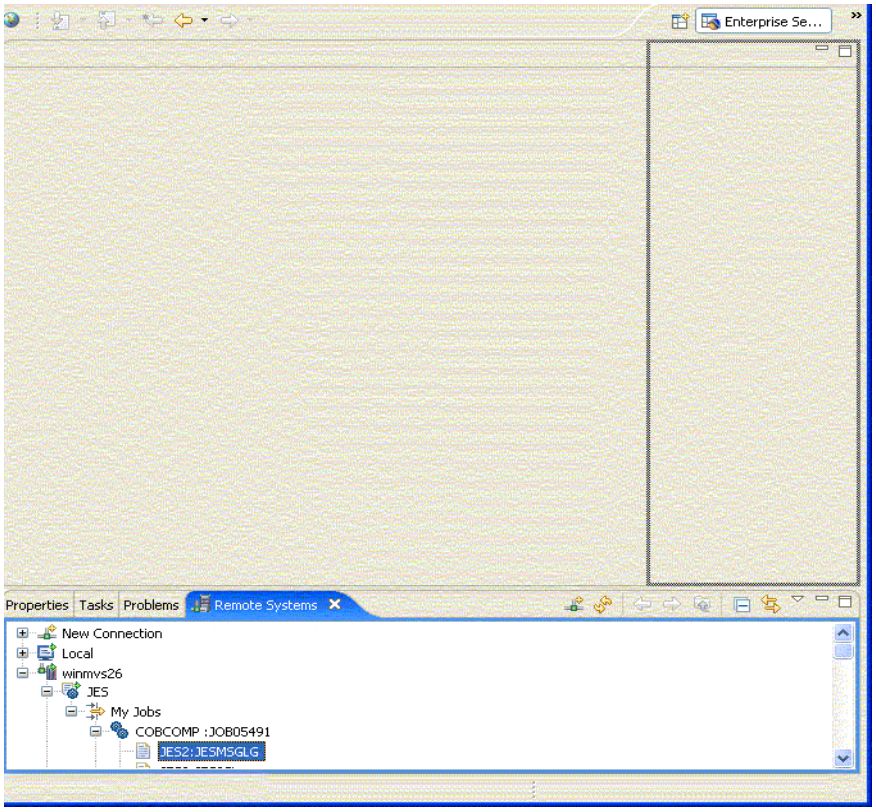
- Click on Show View dropdown



- Select Remote Systems from list

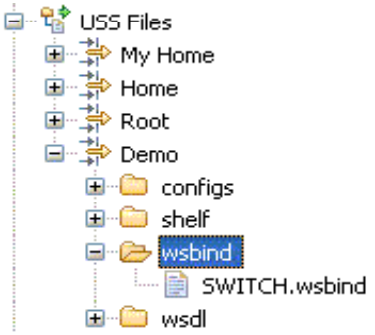


- Move Remote Systems window to make better use of display



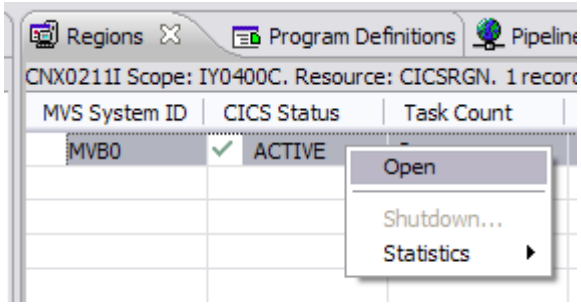
- Drag SWITCH.wsbind from EST Project Explorer window to Remote Systems window

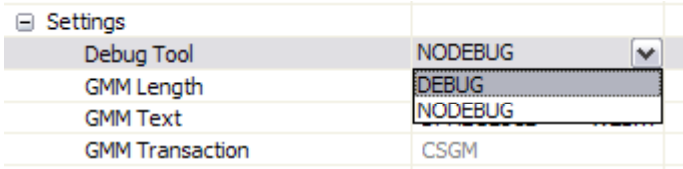
- wsbind file has been copied to USS



Manage using the CICS Explorer

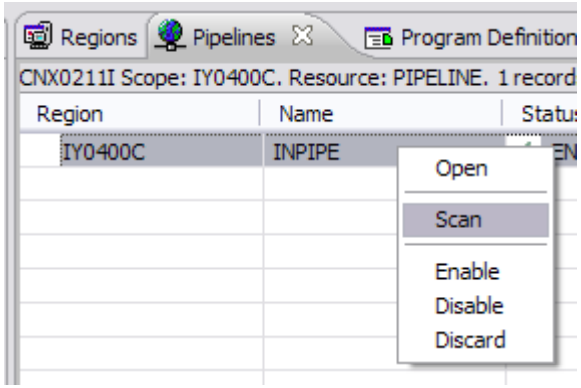
- Select appropriate CICS region



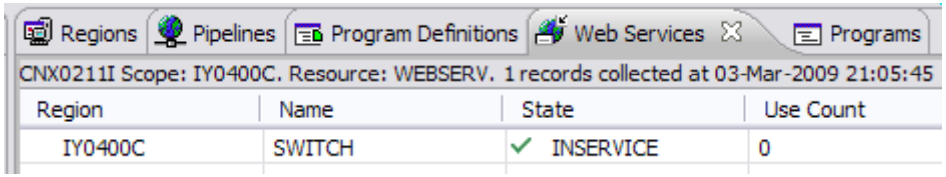


- Change DEBUG status to NODEBUG

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

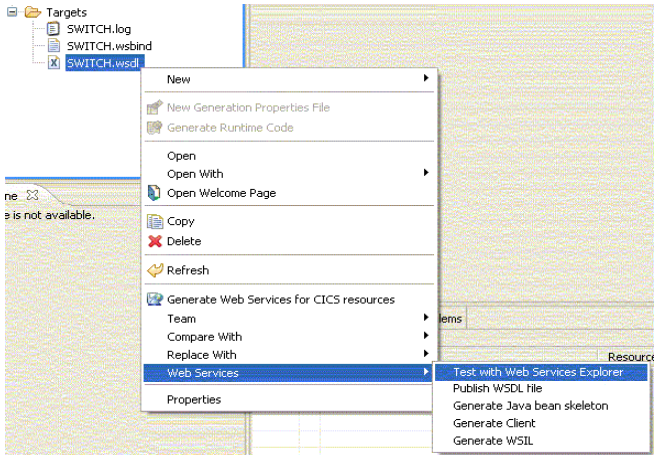


- Select Web Services tab see your deployed Web service

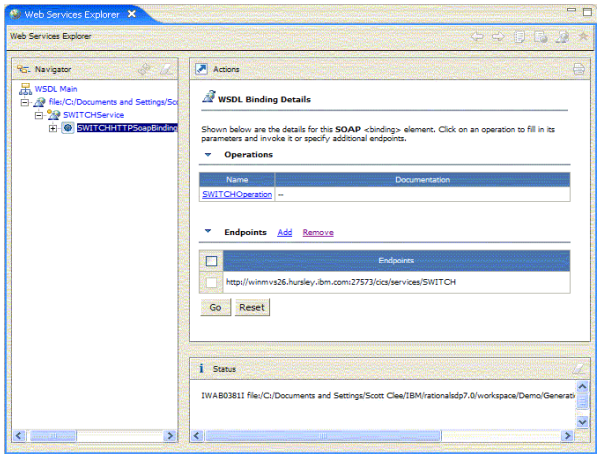


Test the Web service

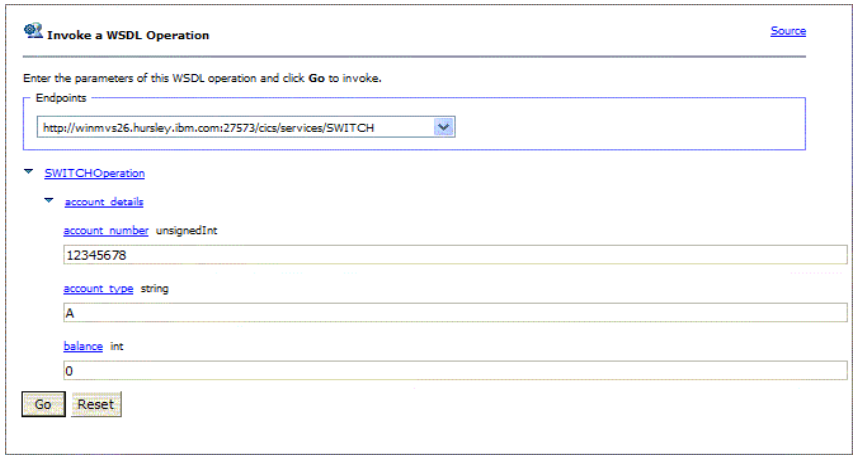
- In Enterprise Service Tools perspective right click on generated WSDL and select Test with Web Services Explorer



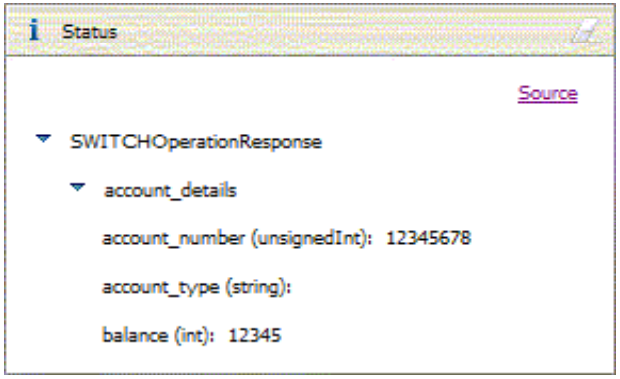
- Select SWITCHOperation from list



- Input data for SOAP request and click Go



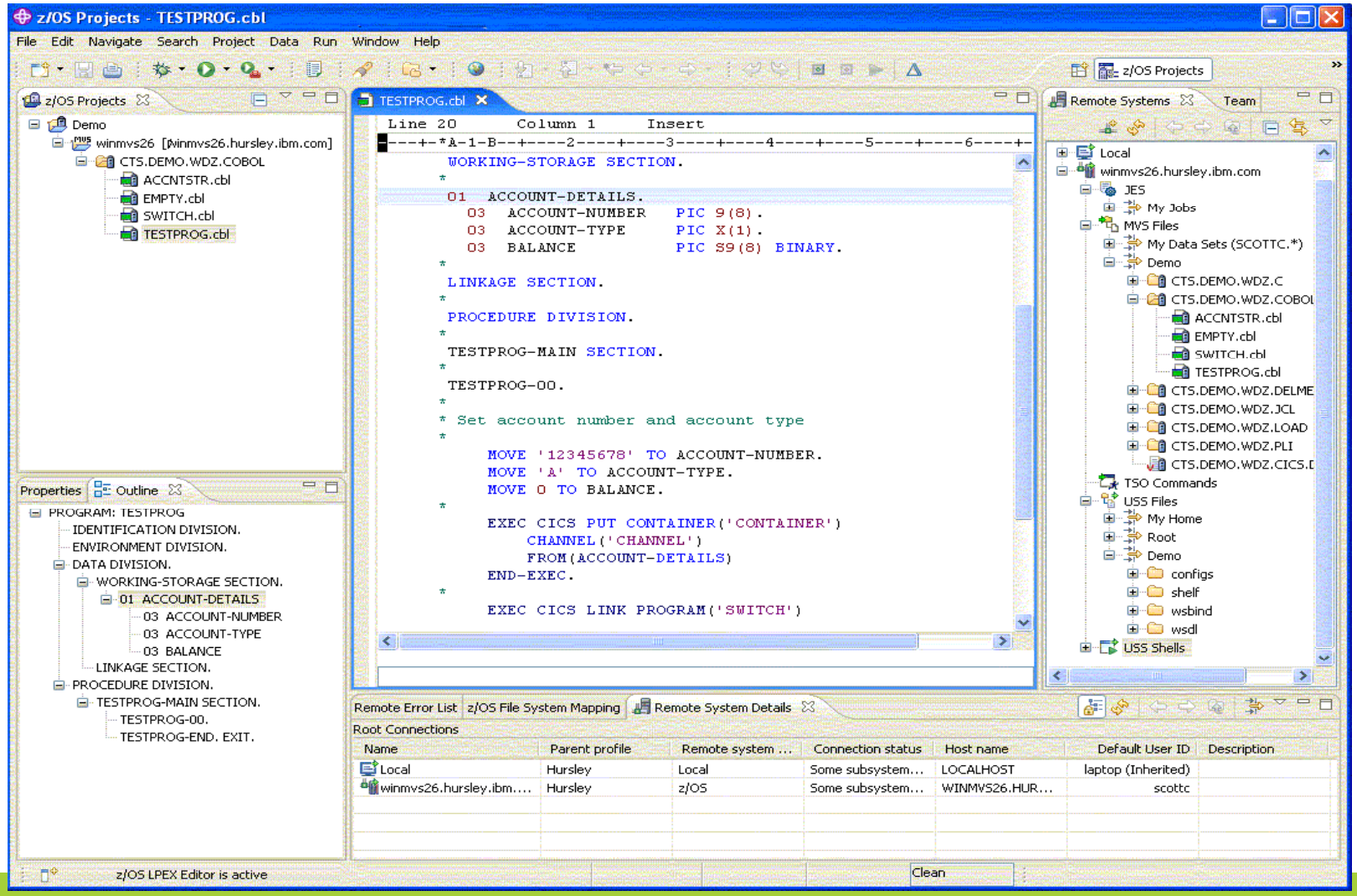
- Observe response data



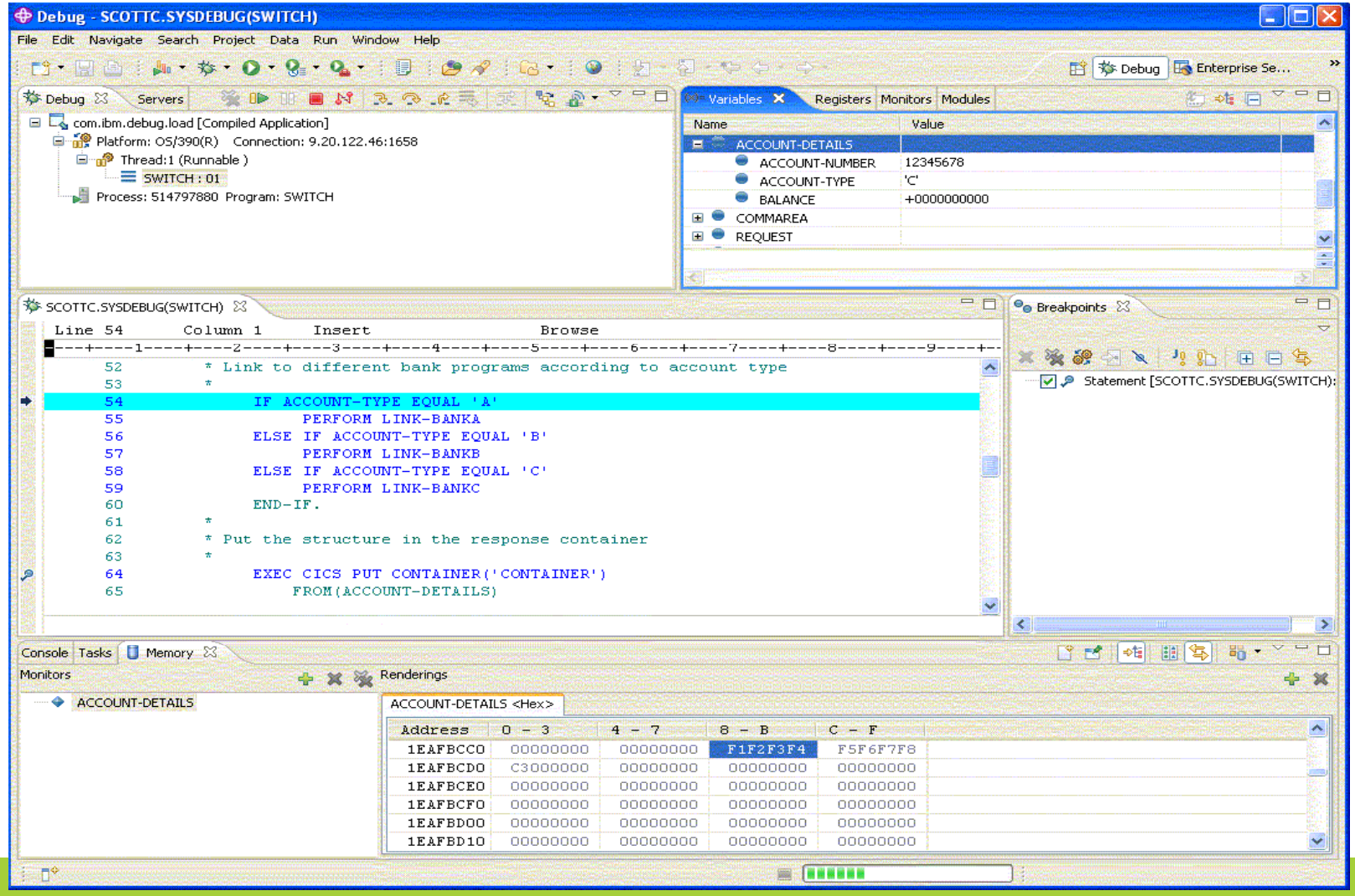
- Click on Source to view request and response XML

RDz screen shots

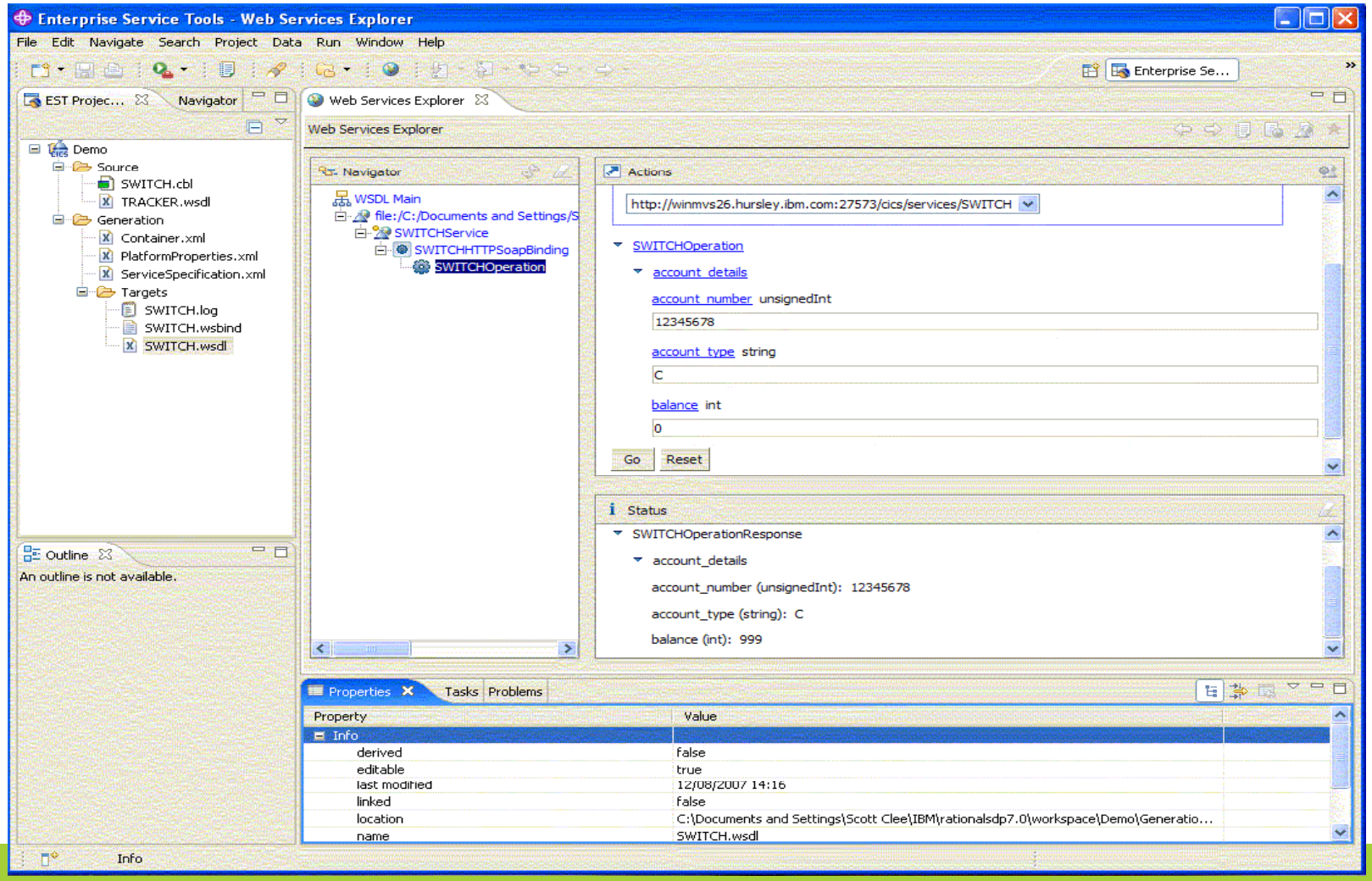
z/OS Projects perspective



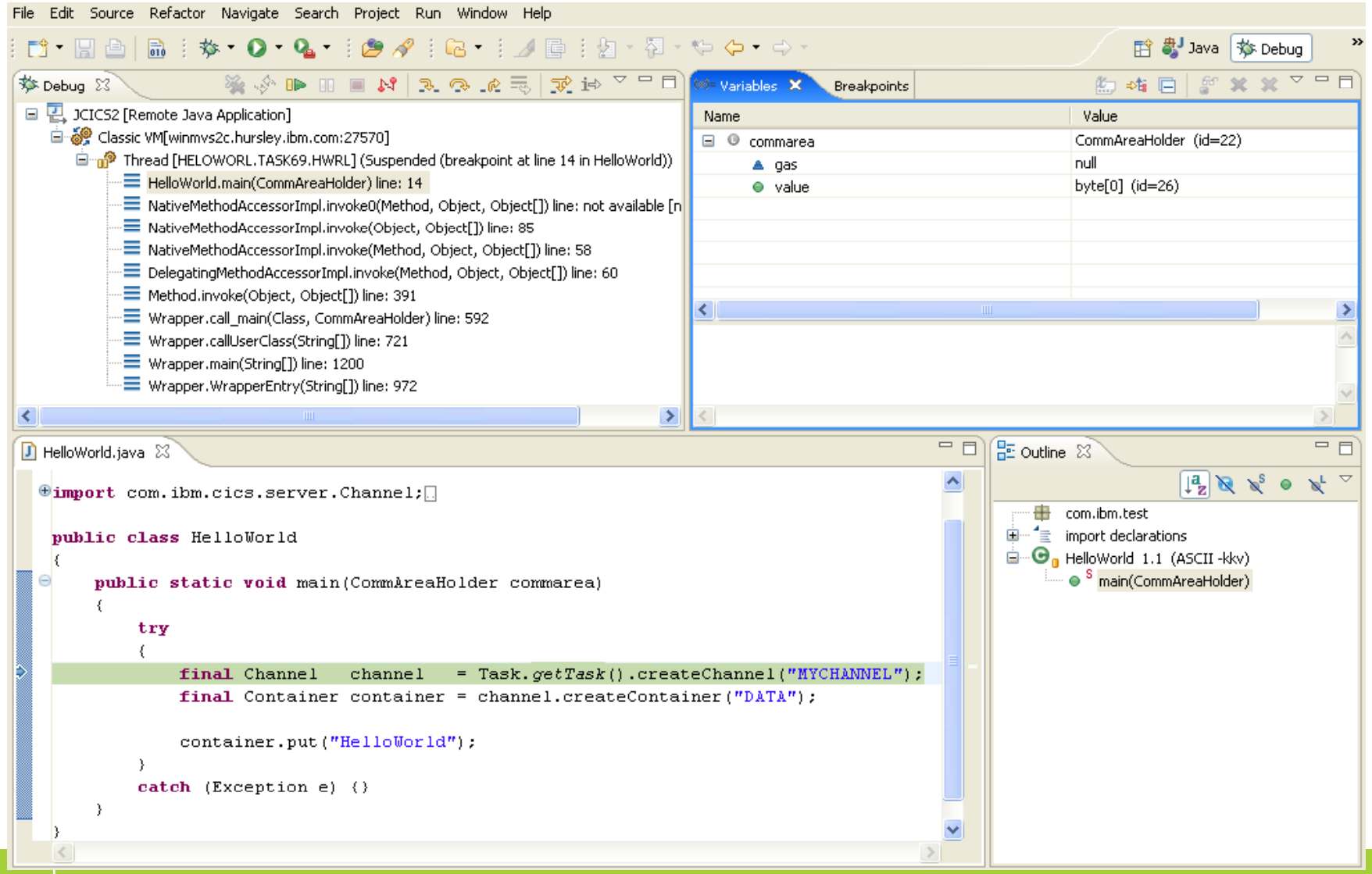
Debug perspective



Enterprise Service Tools perspective



Debugging a CICS Java program



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/