

RDz User/Customer Experience series

ISPW Meta-data + RDz APIs = Accelerated Developer Productivity

Jack Lema
ISPW Benchmark Technologies

August 3, 2010, 4:30 pm – 5:30 pm
Session 7608



SHARE in Boston

Agenda

- Who I am / Who we are
- Why integrate ISPW with RDz?
- Background concepts for ISPW and RDz
- How each benefits from integration
- What have we integrated so far?
- How we have used the RDz APIs
- Benefits & Challenges the APIs present
- My Advice
- Demo?

Who I am

Things I no longer do:

- ◆ MVS sysprog
- ◆ Assembler product developer
- ◆ IT manager

What I do now:

- ◆ Developer working on ISPW distributed solutions:
 - ◆ Eclipse/RDz plug-ins
 - ◆ Browser interface
 - ◆ Web services

Who we are

- ◆ ISPW Benchmark Technologies Ltd. Since 1986
- ◆ One product focus: ISPW
 - ◆ Enterprise software change/configuration management for mainframe and distributed code
 - ◆ Version and release management
 - ◆ Mainframe and distributed deployment and back-out

Customer Examples

ABN Amro Bank, Amsterdam, Netherlands

Bank of Montreal, Toronto

Commerce Bank, Kansas City MO

FedEx, Memphis, TN

Insurance Corporation of British Columbia, Vancouver

Isracard, Tel Aviv, Israel

Southern California Edison, Los Angeles CA

St George Bank, Sydney Australia

State of California Franchise Tax Board, Sacramento CA

Unum Insurance, Chattanooga TN

What is RDz? ISPW?

IBM Rational Developer for System z supports CICS, IMS, DB2, and WAS while helping make multi-platform application development faster, more efficient, and less costly. RDz accelerates the development of your Mainframe applications, including traditional COBOL, PL/I, C/C++, and Assembler applications, Web services, and XML-based interfaces . RDz is built upon the Eclipse platform.

Integrated Software Parts Wherever (ISPW) is a cross-platform SCM solution, which means it can manage mainframe and distributed components in one seamless solution to analyze change impacts, update and test, and approve and deploy changes. In addition to a comprehensive ISPF interface it includes an Eclipse plug-in and Web browser interfaces.

Why Integrate them?

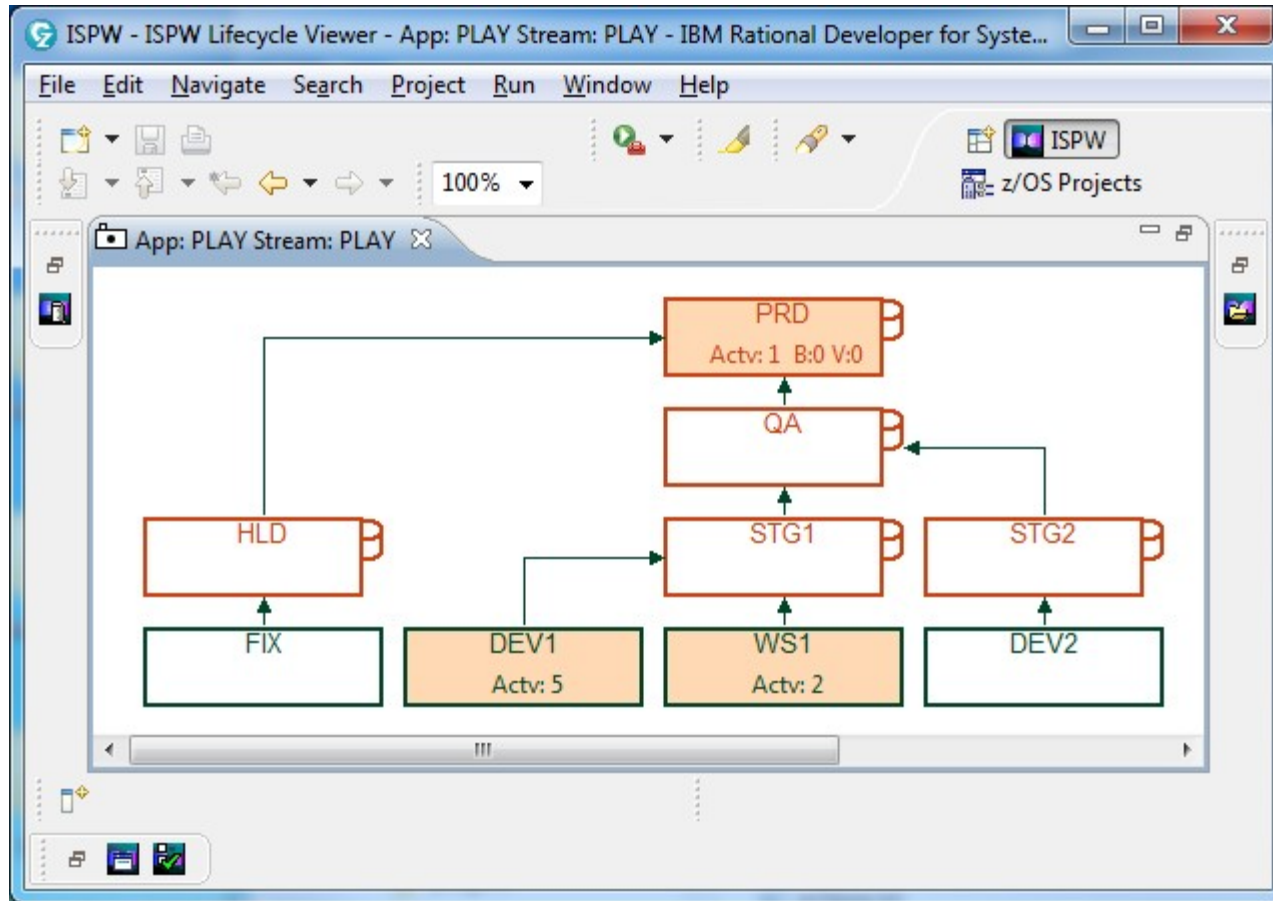
- ◆ Seemed natural since RDz acts upon artifacts ISPW manages
- ◆ ISPW and RDz complement each other nicely
- ◆ ISPW customers want it
- ◆ ISPW stores useful process meta-data which enhances RDz:
 - ◆ Site specific meta-data in DB2 repository
 - ◆ ISPW data about applications is comprehensive
- ◆ ISPW 4.2 and RDz 7.6 have all the pieces in place

ISPW Meta-data Appropriate for RDz

- Attributes which are a function of life-cycle level
 - Source code location
 - syslib/maclib concatenation list and order
- Module type (e.g. cobol, rexx, assembler)
- Compile/link options
- Parameters customized to site standards

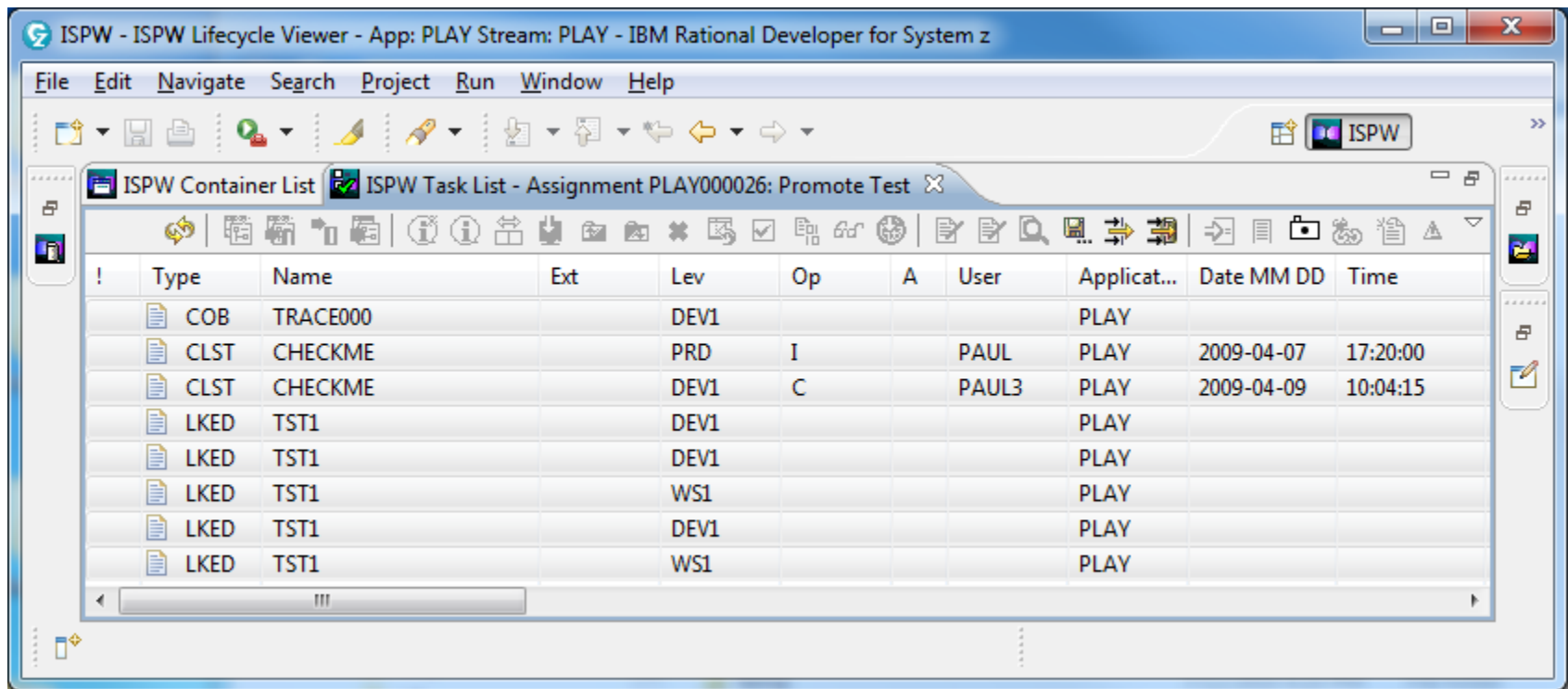
Background concepts (1 of 5)

ISPW Life-cycle: stages from development to production:



Background concepts (2 of 5)

ISPW Task List: organizes modules/versions to upgrade:

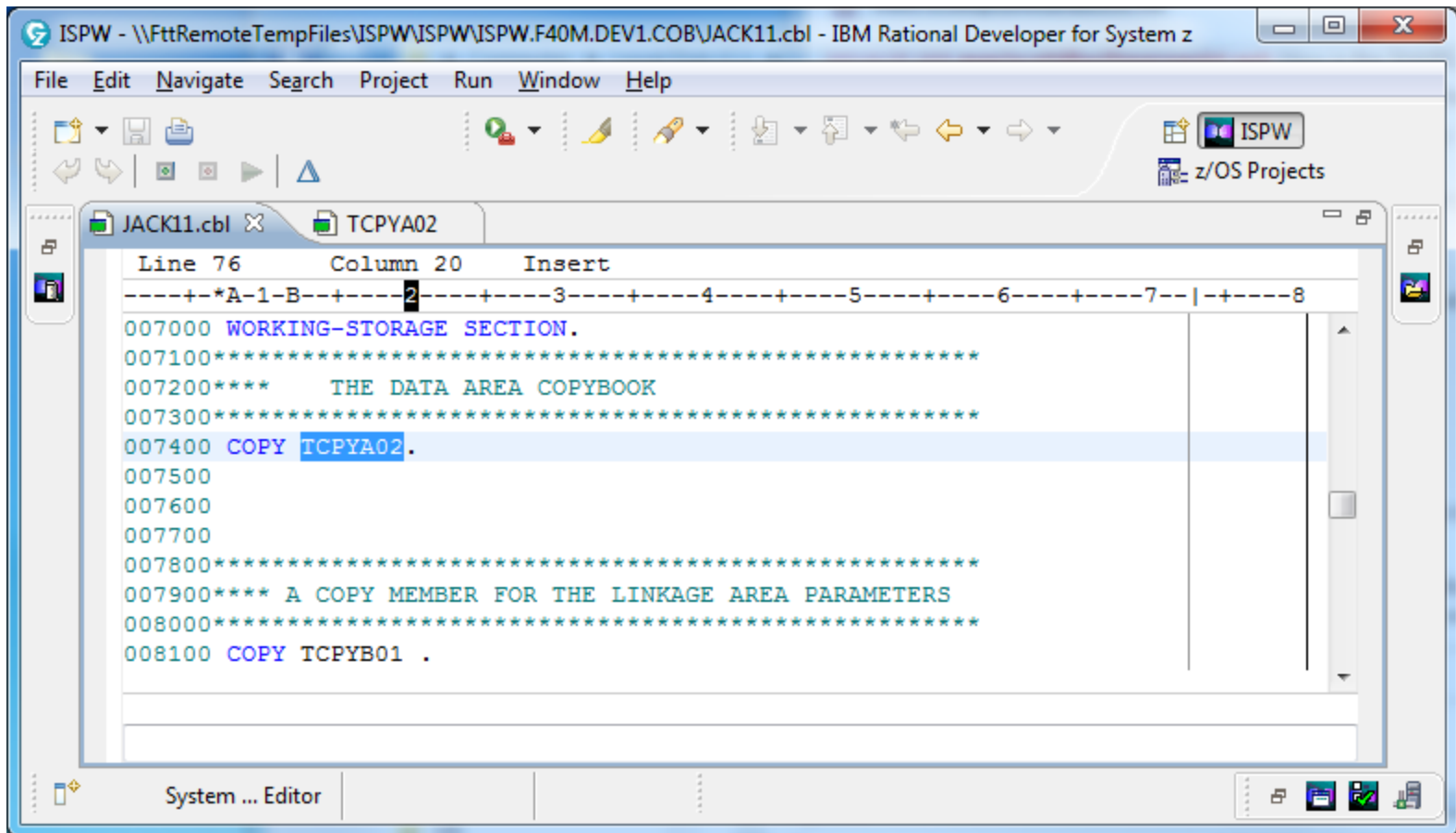


The screenshot shows the 'ISPW Task List - Assignment PLAY000026: Promote Test' window. The table below represents the data shown in the application.

!	Type	Name	Ext	Lev	Op	A	User	Applicat...	Date MM DD	Time
	COB	TRACE000		DEV1				PLAY		
	CLST	CHECKME		PRD	I		PAUL	PLAY	2009-04-07	17:20:00
	CLST	CHECKME		DEV1	C		PAUL3	PLAY	2009-04-09	10:04:15
	LKED	TST1		DEV1				PLAY		
	LKED	TST1		DEV1				PLAY		
	LKED	TST1		WS1				PLAY		
	LKED	TST1		DEV1				PLAY		
	LKED	TST1		WS1				PLAY		

Background concepts (3 of 5)

- RDz LPEX Editor: colorization, syntax check, copylib awareness:

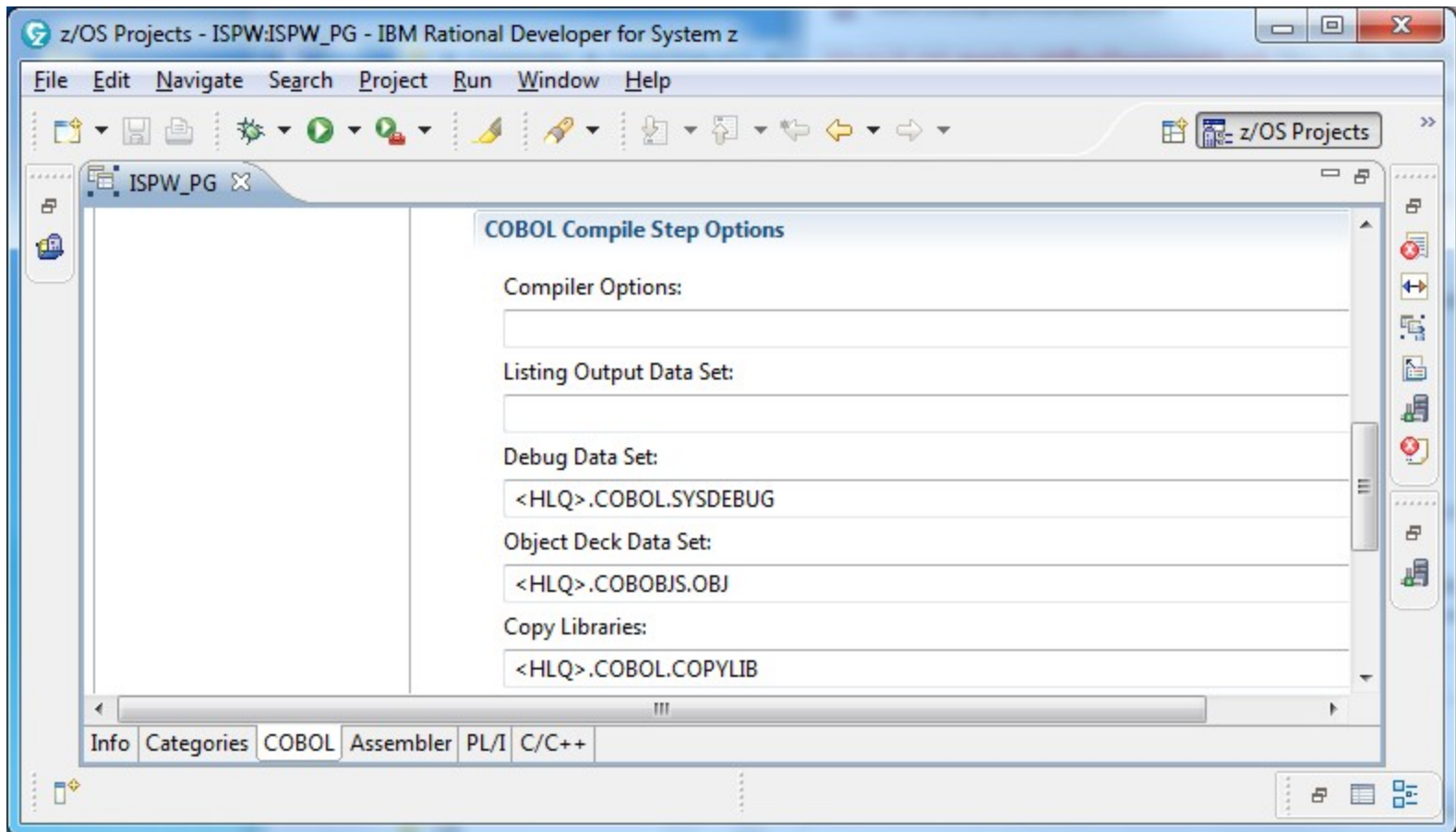


The screenshot displays the IBM Rational Developer for System z (RDz) LPEX Editor interface. The window title is "ISPW - \\FttRemoteTempFiles\ISPW\ISPW\ISPW.F40M.DEV1.COB\JACK11.cbl - IBM Rational Developer for System z". The menu bar includes File, Edit, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations, editing, and navigation. The editor shows two tabs: "JACK11.cbl" and "TCPYA02". The main editing area displays COBOL code with syntax highlighting. A ruler at the top indicates line and column positions. The code includes comments and a COPY statement for TCPYA02.

```
Line 76      Column 20      Insert
-----+*A-1-B-+-----2-----3-----4-----5-----6-----7--|-----8
007000 WORKING-STORAGE SECTION.
007100*****
007200****      THE DATA AREA COPYBOOK
007300*****
007400 COPY TCPYA02.
007500
007600
007700
007800*****
007900**** A COPY MEMBER FOR THE LINKAGE AREA PARAMETERS
008000*****
008100 COPY TCPYB01 .
```

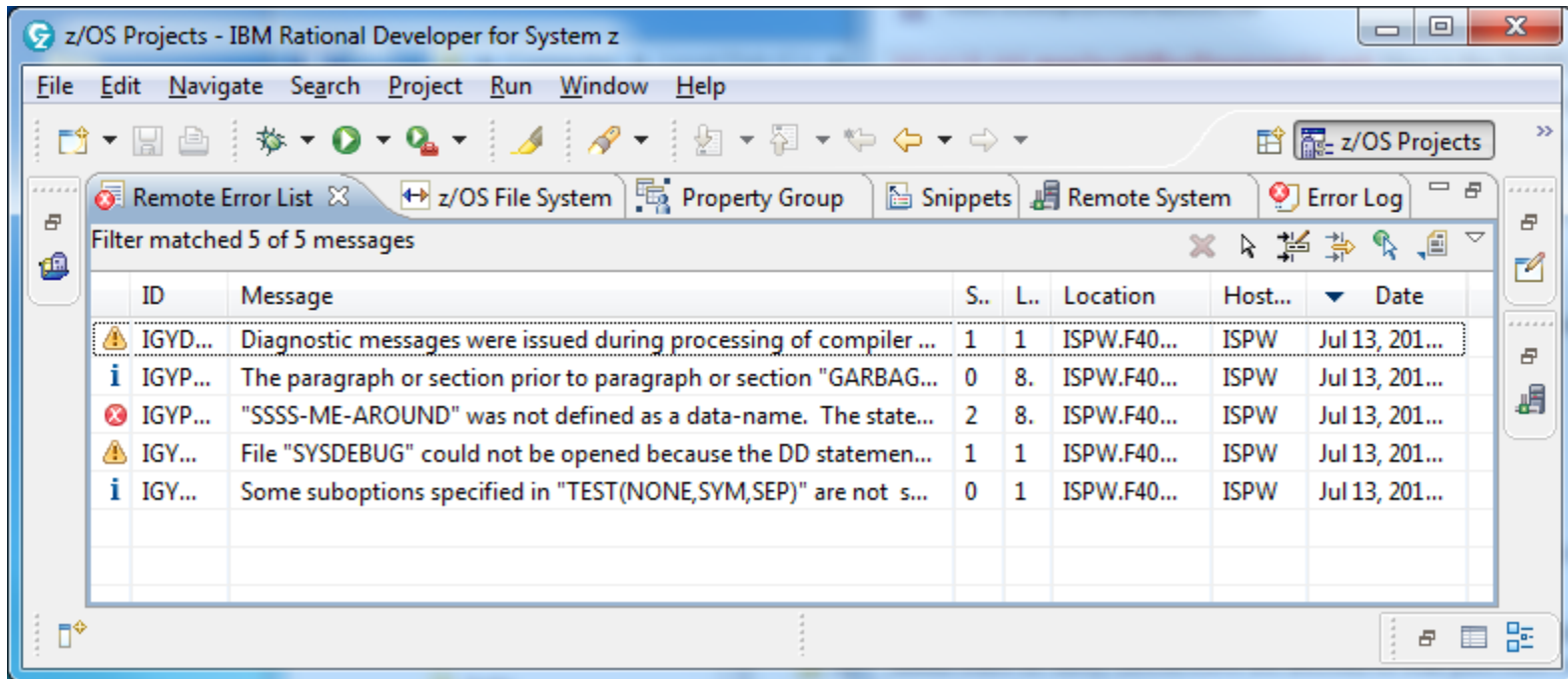
Background concepts (4 of 5)

Property Groups: RDz meta-data container



Background concepts (5 of 5)

RDz Remote Error List: linked to source lines

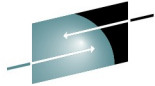


How RDz benefits from ISPW

- ISPW adds SCM features to RDz
- ISPW tells RDz where the source code and copylibs for a module at a given life-cycle level reside.
- ISPW organizes a programmer's tasks into assignments
- ISPW “promote” performs level-appropriate deployment
- ISPW can provide application-specific compiler options

How ISPW benefits from RDz

- LPEX is a wonderful editor of mainframe source (ISPW has no editor) complete with ENQ and recovery support
- RDz offers debug capability, of which ISPW has none
- RDz generates code which ISPW can then manage
- The RDz Remote Error List is a productive way to drive LPEX



SHARE
Technology • Connections • Results

RDz and ISPW Composite Views

z/OS Projects - TSUBR31.cbl - IBM Rational Developer for System z

File Edit Navigate Search Project Data WD/z Samples Troubleshooting Run Window Help

z/OS Projects

- Demo Project
 - Demo Subproject for DALLAS [Dallas]
 - TRNG.PLAY.FIX.COPY(TCPYA32)
 - TRNG.PLAY.FIX.COPY(TCPYB32)
 - TRNG.PLAY.FIX.COB(TSUBR31).cbl
 - TRNG.PLAY.FIX.COB(TSUBR32).cbl
 - Proj1

TSUBR31.cbl

```

Line 70      Column 29      Insert
*A-1-B-----2-----3-----4-----5-----6-----7-----8
005800 LINKAGE SECTION.
005900 *****
006000 **** A COPY MEMBER FOR THE LINKAGE AREA ****
006100 *****
006200 COPY TCPYB31.
006300
006400 PROCEDURE DIVISION USING PASS-ME-AROUND.
006500
006600 00000-MAIN-PROCEDURE.
006610
006700 IF NOT INFILE-OPEN THEN
006800 MOVE 'Y' TO INFILE-OPEN-FLAG
006810 MOVE 'N' TO INFILE-EOF-FLAG
006900 OPEN INPUT INFILE.
007000
007001 MOVE SPACES TO INCOMING-DATA.
007002
007100 IF ACTION-READ THEN
007200 PERFORM GET-ME-A-RECORD THRU GET-ME-A-RECORD-X
007300 UNTIL INFILE-EOF OR KEEPER.
007400
007700 IF ACTION-CLOSE THEN
  
```

Remote Error List: Remote System Details | ISPW Container List | **ISPW Task List** | ISPW Component Version List | ISPW Repository Explorer | z/OS File System Mapping

ISPW Task List - Assignment PLAY000178: SHARE DEMO

I	Type	Name	Ext	Lev	Op	A	User	Appli...	Date M...	Time	Status	Message	iVer	bVer	rVer	Env
	COB	TSUBR31		FIX	C		CLIVE	PLAY	2009-02-24	18:32:21			0	3	3	TEST
	COB	TSUBR32		FIX	C		CLIVE	PLAY	2009-02-24	18:32:17			0	4	4	TEST
	COPY	TCPYA32		FIX	C		CLIVE	PLAY	2009-02-24	18:29:08			0	5	5	TEST
	COPY	TCPYB32		FIX	C		CLIVE	PLAY	2009-02-24	18:28:57			0	5	5	TEST
	COB	VLECOB01		FIX	C		WX0...	PLAY	2009-01-26	19:15:35		OK	0	0	0	TEST
+	COB	TCPYA10		FIX	C		CLIVE	PLAY	2009-01-23	00:16:37	Check Versions;		0	3	3	TEST
	COPY	TCPYA33		FIX				PLAY					0	0	0	OUTS
	COPY	TCPYA31		FIX				PLAY					0	0	0	OUTS
	COPY	TCPYA15		FIX				PLAY					0	0	0	OUTS
	COPY	TCPYA14		FIX				PLAY					0	0	0	OUTS
	COPY	TCPYA13		FIX				PLAY					0	0	0	OUTS
	COPY	TCPYA12		FIX				PLAY					0	0	0	OUTS
	COB	TSUBR33		FIX				PLAY					0	0	0	OUTS
	COB	TSUBR33		FIX				PLAY					0	0	0	OUTS

Properties Outline

- PROGRAM: TSUBR31
 - IDENTIFICATION DIVISION.
 - ENVIRONMENT DIVISION.
 - DATA DIVISION.
 - FILE SECTION.
 - FD INFILE
 - WORKING-STORAGE SECTION.
 - LINKAGE SECTION.
 - PROCEDURE DIVISION
 - 00000-MAIN-PROCEDURE.
 - GET-ME-A-RECORD.
 - GET-ME-A-RECORD-X.

Total items: 14 (Filters: level=FIX)

What have we done so far?

- ◆ Launch LPEX from the ISPW task list familiar to our customers
- ◆ Populate the Remote Error List from an ISPW “generate”
- ◆ Add source file links to MVS sub-project of z/OS project

Launch LPEX from the ISPW task list

- ◆ Create a single, reusable property group with appropriate categories (Cobol, Assembler, JCL, etc.)
- ◆ Set default properties
- ◆ Associate the PG with the resource (PDS member)
- ◆ Set a property value override for members who differ from the default: PDS/member, syslib concatenation
- ◆ Launch LPEX against that resource

This single PG can be associated with multiple resources as long as they are on the same system.

Populate the Remote Error List

- ◆ Retrieve compiler error XML file (via RDz)
 - Obtain a reference to the XML file
 - Ask RDz to download the XML file

- ◆ Call our custom XML parser to extract the data:
 - Build resource ref for source/copylib members
 - Parse error details into entry (aka Marker)
 - Add source file location for LPEX
 - Add marker to Remote Error List for each error

Benefits of the RDz APIs

Allow injection of Property Group overrides (e.g. syslibs) to influence many RDz actions

Allow management of z/OS project resources

Provide access to file transport functions (including JES)

Enable use of LPEX with ENQ protection and recovery

Provide complete control over the Remote Error List

Challenges of Using the RDz APIs

Early documentation was non-existent (since corrected)

No road map of how to combine methods to achieve a particular objective

Project View and Remote System Explorer treat resources a little differently

These APIs may not be stable yet.

My Advice

Know Java and Eclipse plug-ins before you start

Use RDz itself to create any RDz plug-ins

Study the API documentation

Study the sample code that comes with RDz

Use Eclipse “Content Assist” to explore available methods

Make friends at IBM: start with the gentlemen here from IBM

Next phase of integration



Analyze work flows to optimize “ISPW driving RDz”

Fill in some known gaps in our current offering

Explore additional uses for ISPW meta-data

Develop “RDz driving ISPW” features

Summary



Combining the strengths of RDz and ISPW has resulted in a better mainframe application development process from the workstation:

Fewer points of failure through automation

Reduced learning curve for new (mainframe) developers

Ability to develop mainframe apps on platform of choice

More success ... more often

For more information:



RDz:

Overview and trial download:

<http://www-01.ibm.com/software/awdtools/rdz/>

DeveloperWorks Page:

<http://www.ibm.com/developerworks/rational/products/rdz/>

ISPW: jack.lema@ispw.com

How ISPW Works (3 minute video)

http://ispw.com/workinprogress/How_ISPW_works/

ISPW web site:

<http://www.ispw.com/>

Demo (if time permits)



Invoking LPEX from the ISPW Task List

Populating the Remote Error List and showing it in action

Adding a resource to an MVS Sub-project from the Task List