IBM Software Group

IBM Developer for z Systems – for ISPF Developers

Module 5 – Remote Systems – MVS Dataset Access & Organization

DevOps

Jon Sayles/IBM: jsayles@us.ibm.com

IBM Trademarks and Copyrights

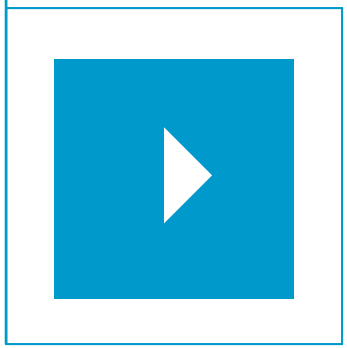
The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials 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 these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials 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.

IBM, the IBM logo, the on-demand business logo, Rational, the Rational logo, and other IBM Rational products and services are trademarks or registered trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.

UNIT

The IDz Workbench



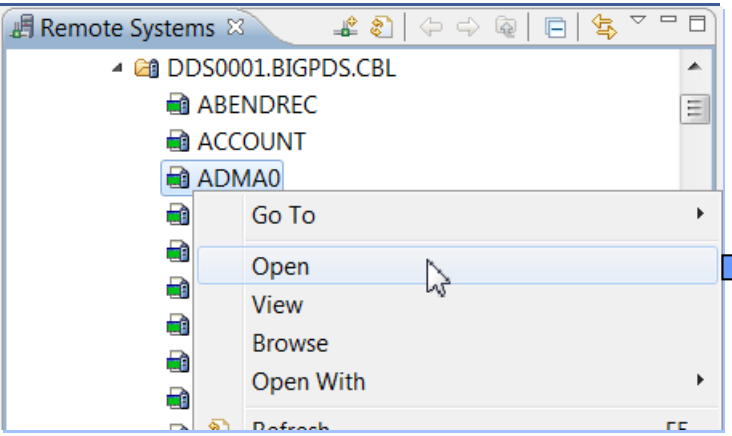
Topics:

- Accessing and Organizing MVS Datasets
- **z/OS File Mapping and Property Groups**
- Miscellaneous Remote Systems Capabilities

Note that if you're using IDz v14.1.3 or later you will not need to do File Mapping.
IDz detects the file type from its contents.

z/OS File Mapping

- In order for IDz to know which editor to open and how to download the contents of an MVS file you need to “map the file”



What software language?

Download as text or binary (EBCDIC)

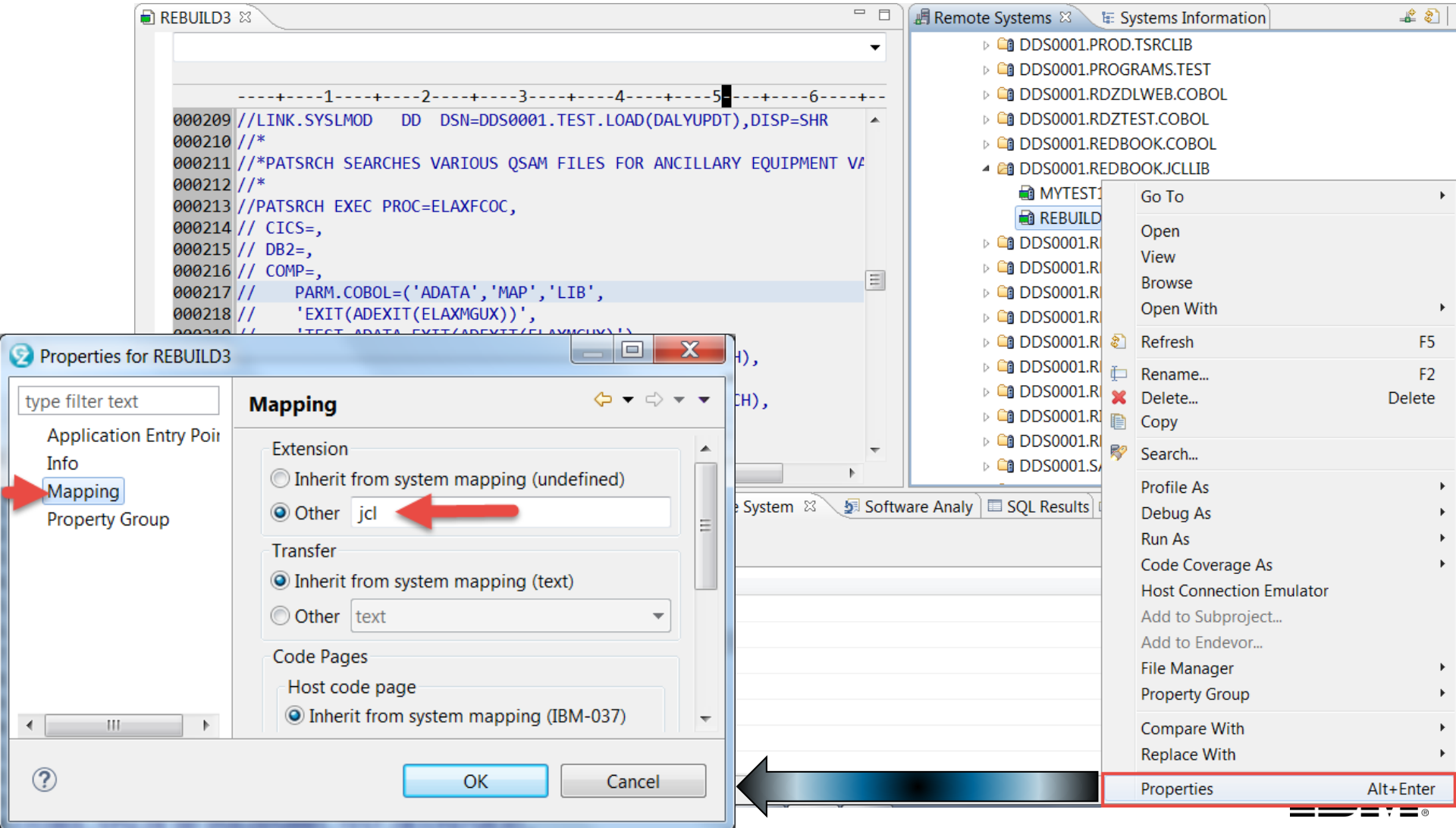
Any specific Code Pages?

- This can be done in one of three ways:
  1. Map individual PDS members
  2. Map an entire PDS
  3. Map every file named a certain way throughout your connection

If you’re using IDz v14.1.3 or later you will not need to do File Mapping, as the product detects the file type from its contents.

1. Map individual PDS Members Using Properties

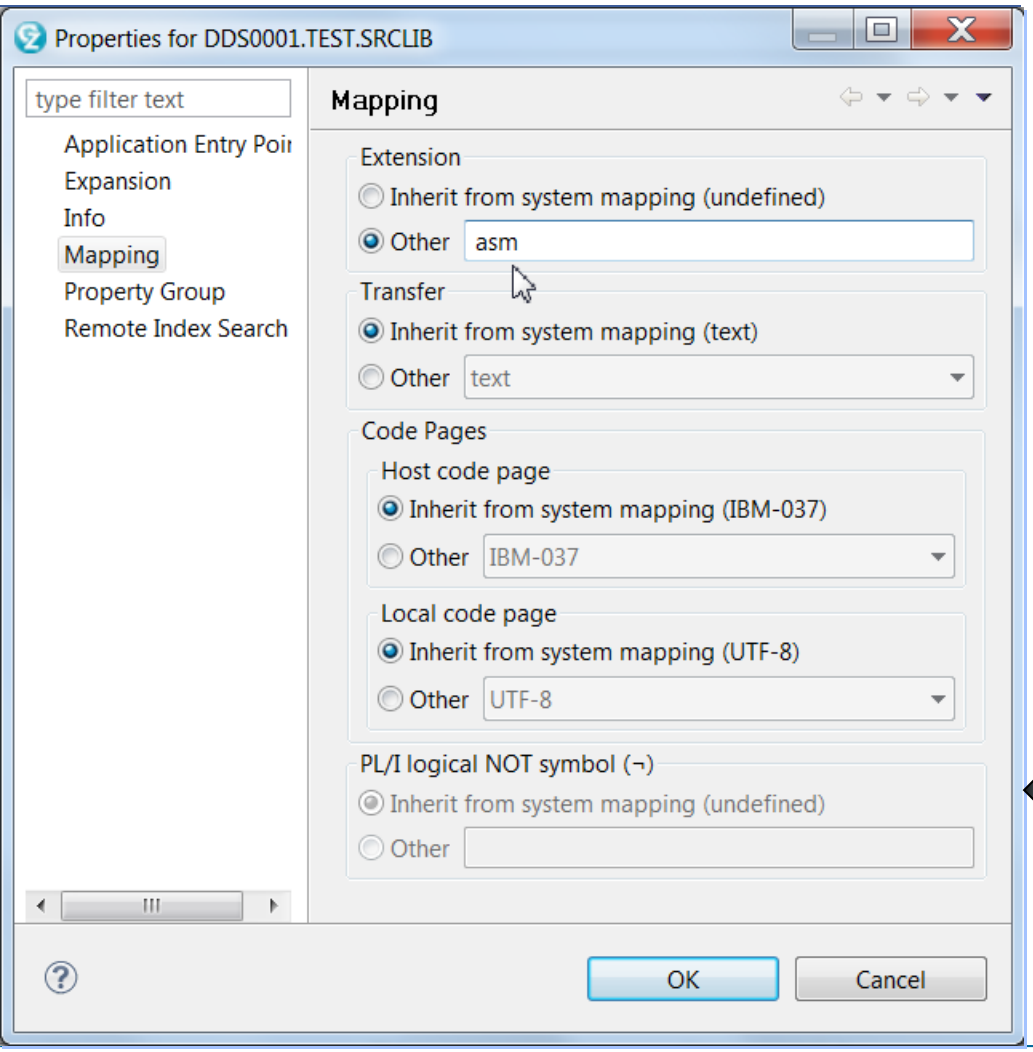
- The simplest / easiest way to map any given PDS member is to: 1. Right-click on the member and from Properties specify the Mapping you want for the file
  - Example: A member of a JCL Library mapped to jcl using Properties

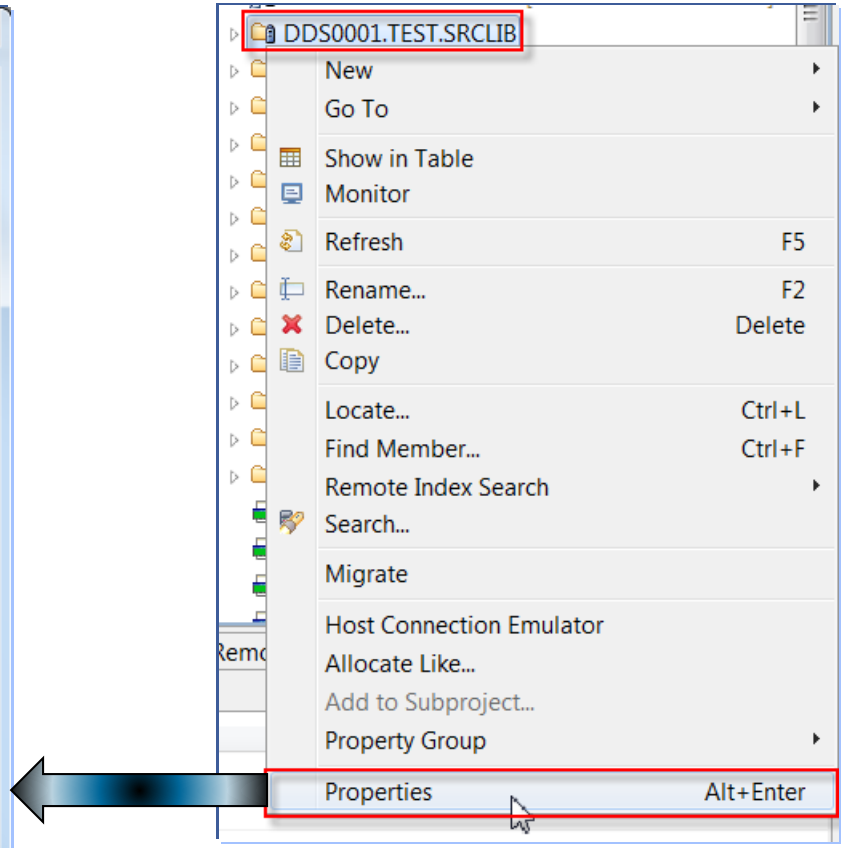


2. Map a PDS (Library) Using Properties

If all the members in a PDS are of the same language type you can map the entire PDS

- **Right-click on the library name in Remote Systems**
- **From Properties specify the Mapping**





3. Map MVS File Names Within a Connection

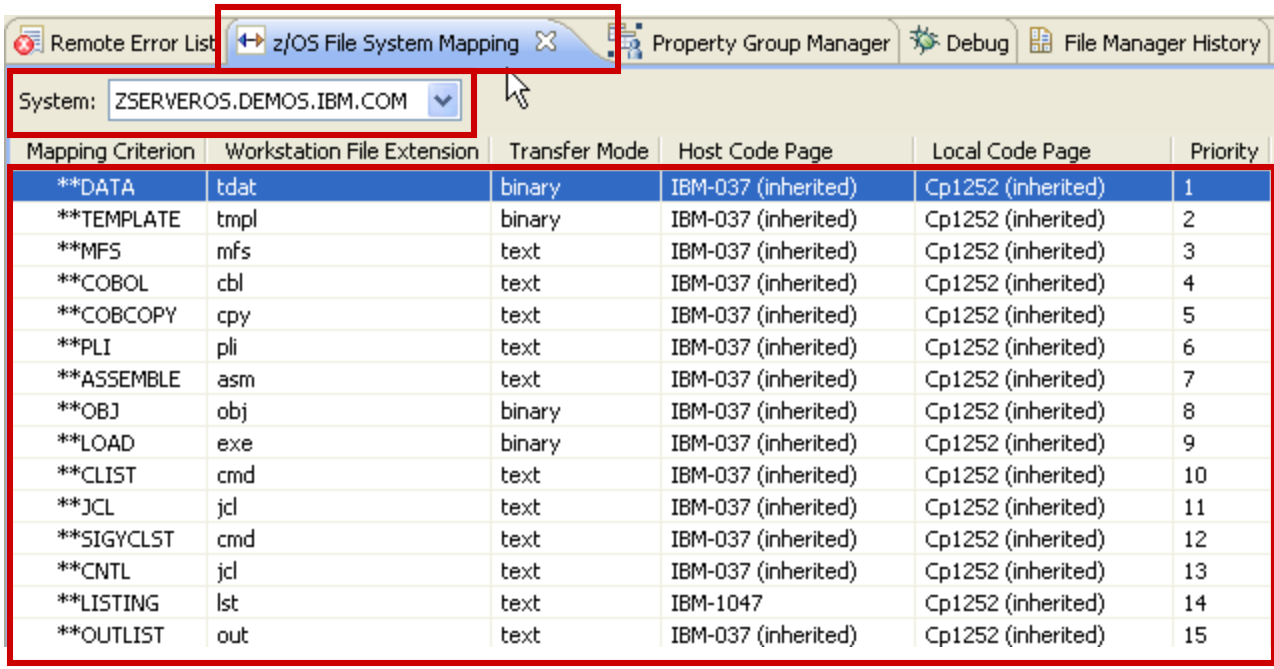
When you define a connection to a remote system, IDz provides a set of ~30 default DSN name patterns mapped to a Workstation File Extension.

- Ex. default mapping associates an MVS dataset ending with COBOL to the .cbl Workstation File Extension.
- All of the default system-wide mappings are listed in the z/OS File System Mapping view.

Z/OS File System Mapping view➔

Connection name➔

Default file mapping specifications➔



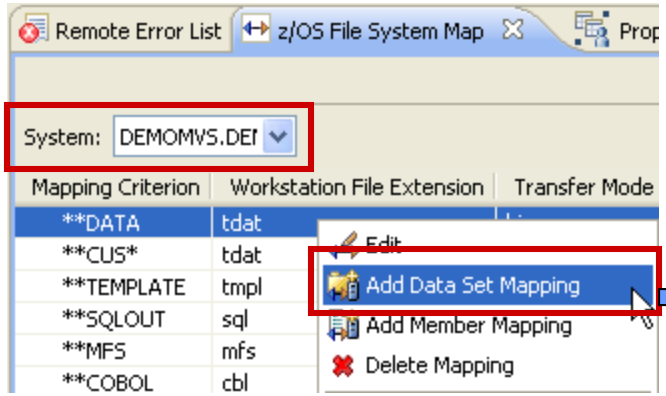
Mapping Criterion	Workstation File Extension	Transfer Mode	Host Code Page	Local Code Page	Priority
**DATA	tdat	binary	IBM-037 (inherited)	Cp1252 (inherited)	1
**TEMPLATE	tmpl	binary	IBM-037 (inherited)	Cp1252 (inherited)	2
**MFS	mfs	text	IBM-037 (inherited)	Cp1252 (inherited)	3
**COBOL	cbl	text	IBM-037 (inherited)	Cp1252 (inherited)	4
**COBCOPY	cpy	text	IBM-037 (inherited)	Cp1252 (inherited)	5
**PLI	pli	text	IBM-037 (inherited)	Cp1252 (inherited)	6
**ASSEMBLE	asm	text	IBM-037 (inherited)	Cp1252 (inherited)	7
**OBJ	obj	binary	IBM-037 (inherited)	Cp1252 (inherited)	8
**LOAD	exe	binary	IBM-037 (inherited)	Cp1252 (inherited)	9
**CLIST	cmd	text	IBM-037 (inherited)	Cp1252 (inherited)	10
**JCL	jcl	text	IBM-037 (inherited)	Cp1252 (inherited)	11
**SIGYCLST	cmd	text	IBM-037 (inherited)	Cp1252 (inherited)	12
**CNTL	jcl	text	IBM-037 (inherited)	Cp1252 (inherited)	13
**LISTING	lst	text	IBM-1047	Cp1252 (inherited)	14
**OUTLIST	out	text	IBM-037 (inherited)	Cp1252 (inherited)	15

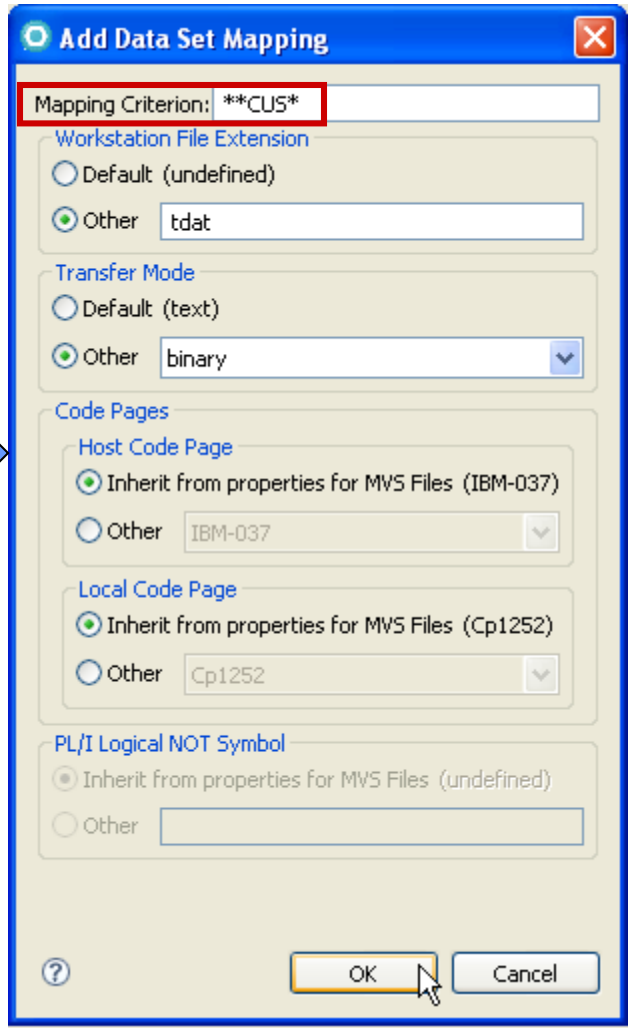
You can customize these mappings to match the naming conventions on your remote system either through the z/OS File System Mapping view or through the Mapping pane in the Properties window.

Set up Custom z/OS File System Mappings for your Datasets – 1 of 2

Steps

1. From the **z/OS File System Map** view
   - From the drop-down, select the System (connection)
   - **Right-Click** and select **Add Data Set Mapping**





2. Specify the mapping characteristics

- Mapping Criterion – for this example:
  - Each double asterisk (**) is a wildcard for a file name level
    - And the double asterisk can also mean “any number of levels”
  - Each single asterisk is a wildcard as part of a dataset name
  - So ****CUS*** - means <anyHLQ>.<any2ndLevelQ>.CUS … w/any suffix
- Workstation file extension (as a file type label)
- The file transfer protocol:
  - **Text** – for ASCII source files
  - **Binary** – for test data datasets

Set up Custom z/OS File System Mappings for your Datasets – 2 of 2

In this example, we have created a Mapping for all datasets named:

<anyHLQ>.

<any2ndLevelQ>.

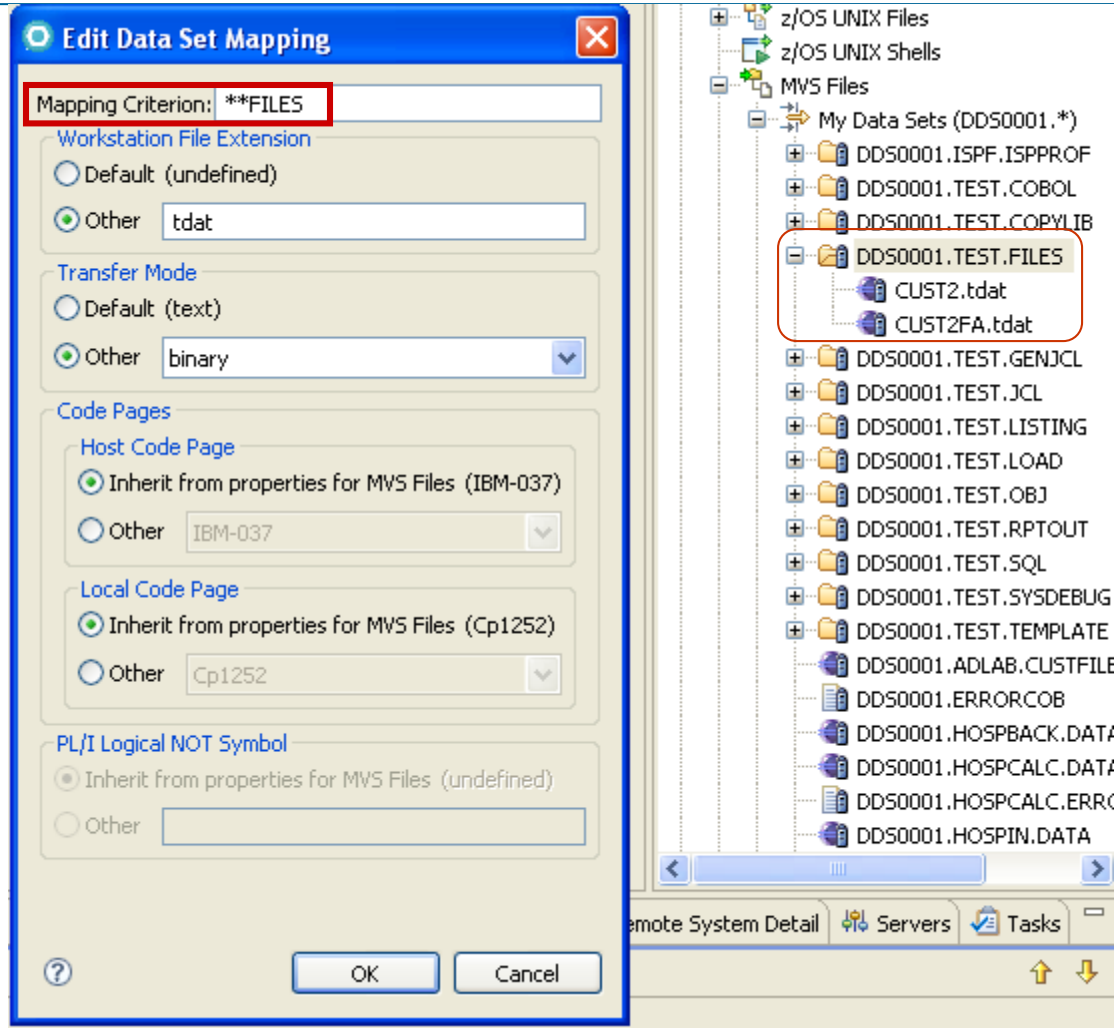
FILES

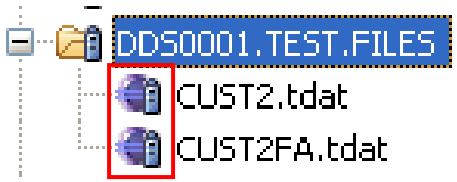
Ex. DDS0001.TEST.FILES

And if the file is a PDS?

- All members inherit the mapping

Don't forget to select your connection (System) before mapping your dataset



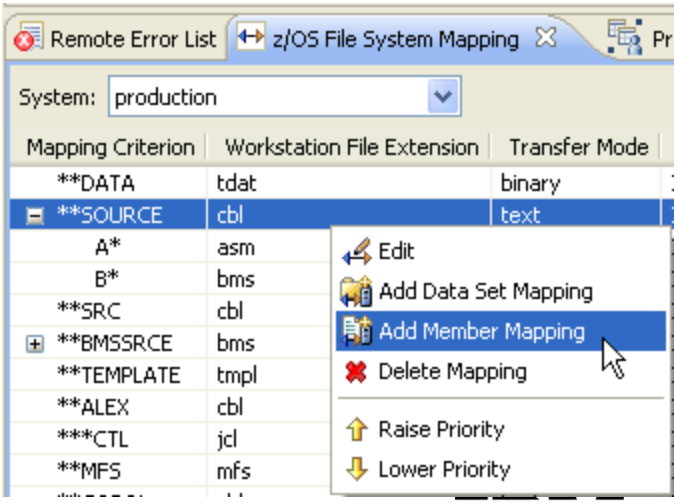


After you successfully add a new z/OS File Mapping for dataset, the default icon associated with file will change

Mapping Members in “Generic” Source Libraries with a Naming Pattern

- You may have generic mainframe source libraries which contain:
  - COBOL
  - Copybooks
  - JCL
  - Assembler
  - BMS
  - MFS
  - etc.

- In order to map disparate member names to the appropriate IDz editor, from **z/OS File System Mapping**
  - Select the **Mapping Criterion**
  - Right-click and select **Add Member Mapping**
  - From the wizard, enter a wildcard string that provides the correct mapping criteria for member names to the associated for their type



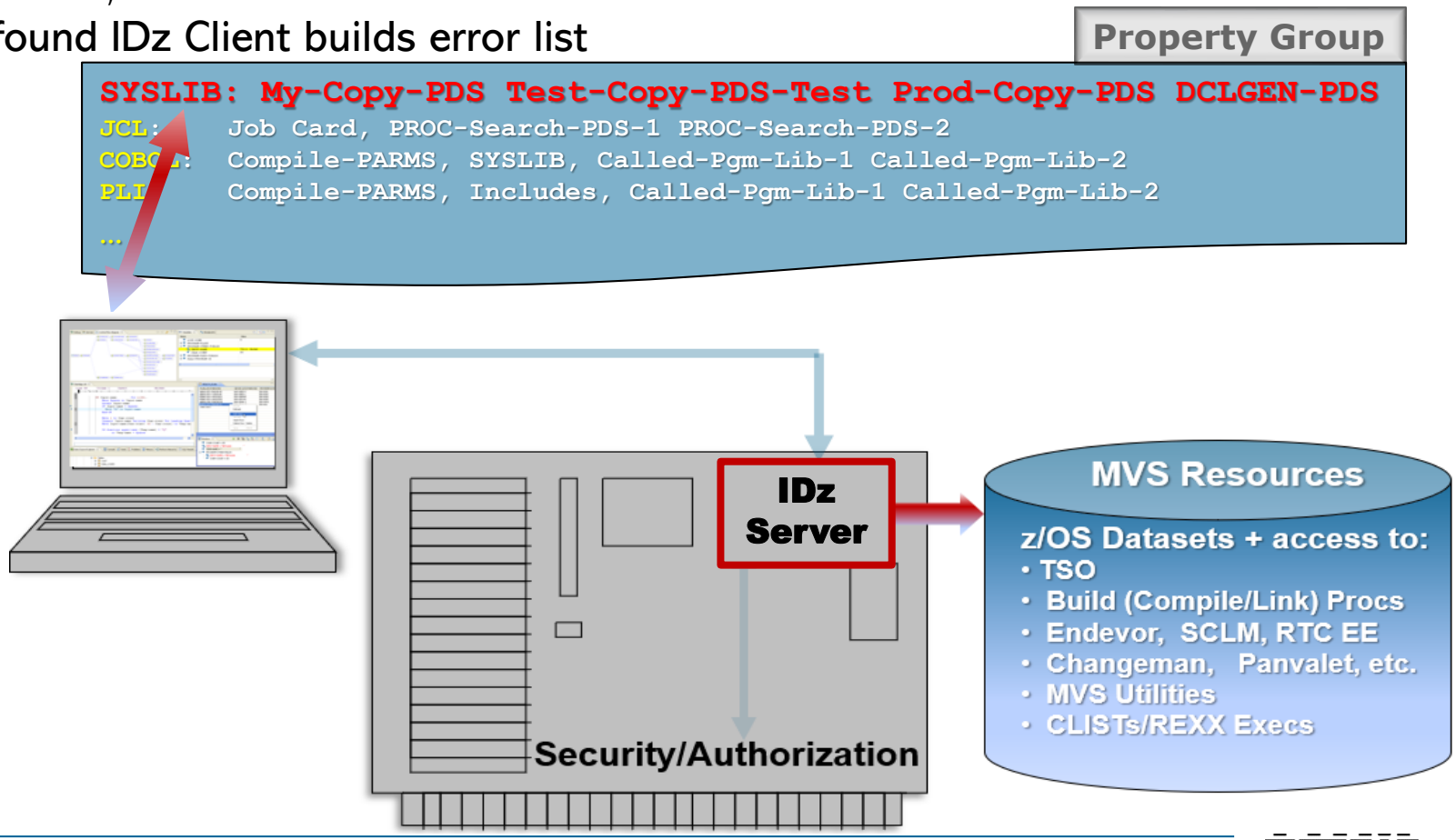
Property Group Usage and Configuration – Considerations

Property Groups are XML files that configure Context Menu actions so that they can be used on your company's LPAR:

- **SYSLIB and Search Library path specification –** for source file editing:
  - Copybook and include dependencies
  - Open/browse called modules
  - JCL PROC lookup
  - Embedded SQL syntax checking
- **Language Compile properties:**
  - Remote Syntax Check
  - Build dependency reports
  - MAP Assembly
- **Integration with 3rd Party products –** that require a Pre-processor:
  - CA-IDMS
  - CA-Telon, Netron-Cap, , CA-Meta Cobol, DTB COBOL, etc.
  - Report Writer
- **DevOps functionality:**
  - zUnit (Unit Test)
  - Code Coverage

Use Case#1 – **Edit Program Source** – Conceptual Product Workflow

1. You open a member in a program source library
2. IDz Server returns program source and parses it for COPY and INCLUDE statements
3. IDz Client passes SYSLIB (PDS Search List) from Property Group to IDz Server – then passes each COPY/INCLUDE Member-Name found in the program source to the Server
4. IDz Server searches PDS list in SYSLIB for each COPY/INCLUDE Member
   - When found, IDz server returns the member to the IDz Client
   - If not found IDz Client builds error list

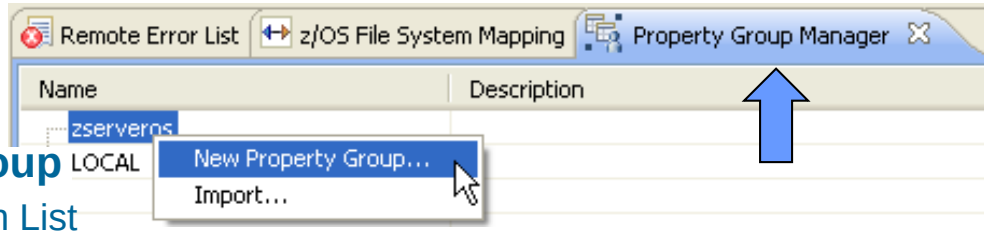


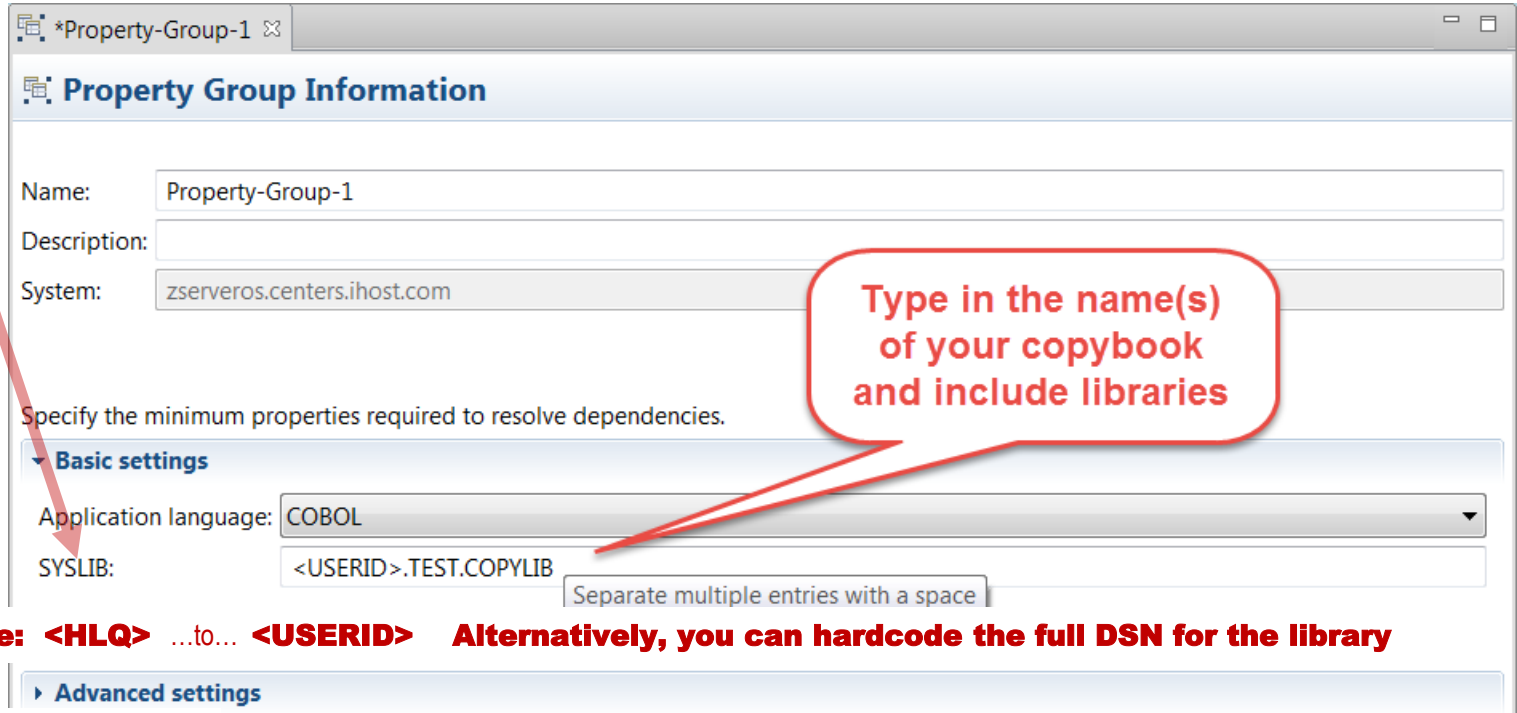
SYSLIB – 1 of 2: **Create the Property Group**

In order for IDz to return COPYBOOK and INCLUDE references found your program source, you must specify SYSLIB *(library search PDS list)*

Steps:

- From the **Property Group Manager** view
  - Select your connection
  - Right-click and select: **New Property Group**
  - **Edit SYSLIB** – specify the ordered PDS Search List





Change: <HLQ> ...to... **<USERID>** **Alternatively, you can hardcode the full DSN for the library**

Press Ctrl+S To save your Property Group edits

/I | JCL | Run | C/C++ | BMS | COBOL | Link | zUnit | Assembler | MFS

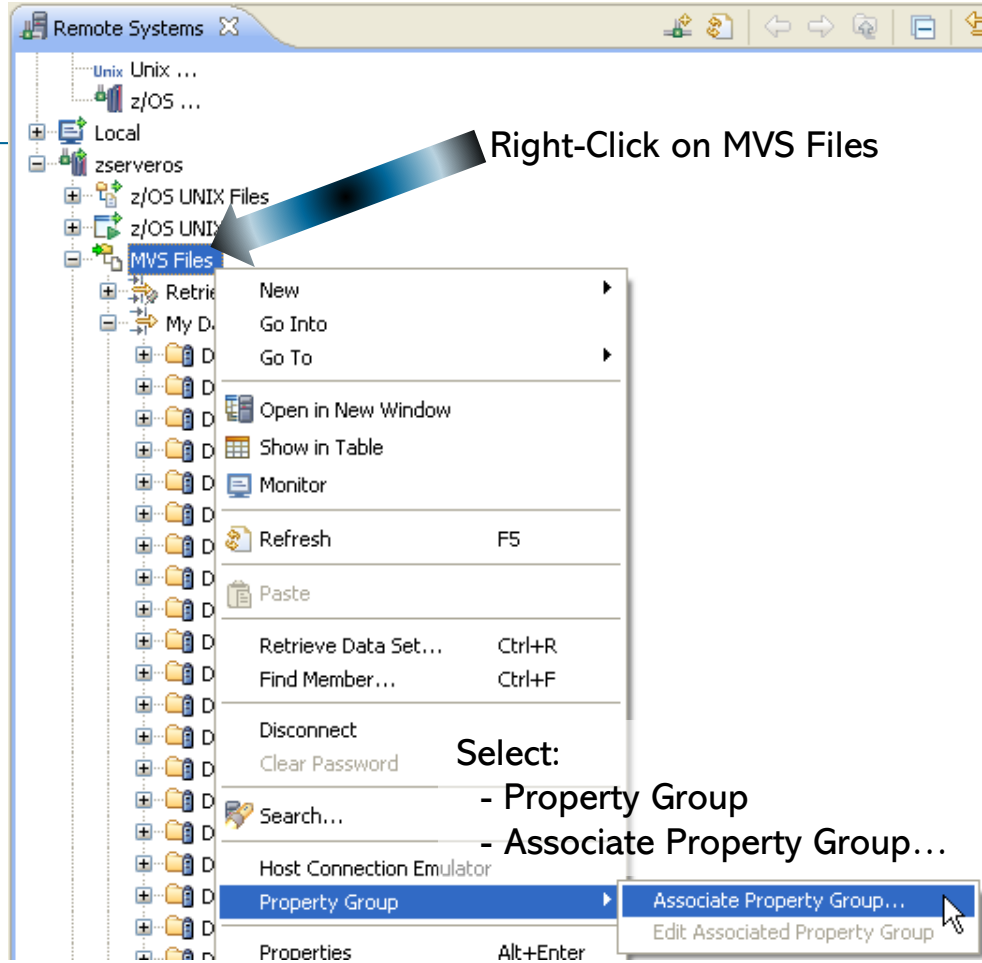
SYSLIB – 2 of 2:

Assign the Property Group

Once your Property Group has been edited, save your edits and assign (Associate) it to MVS Files in Remote Systems

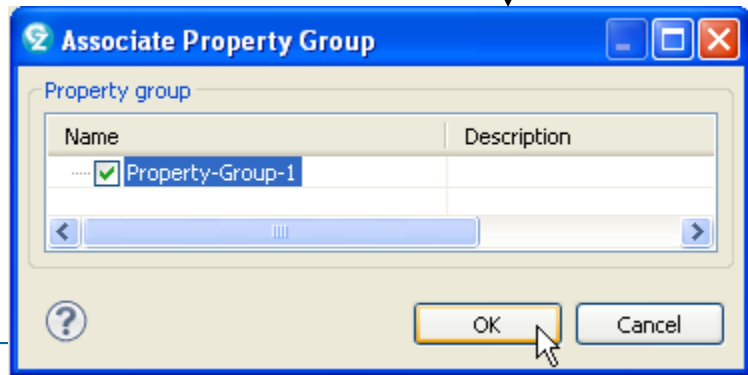
Steps:

- From Remote Systems:
  - Right-click over MVS Files
  - Select Property Group > Associate Property Group…
  - Un-check and Check your named Property Group
  - Click OK



Test your work:

- Open <USERID>.TEST.COBOL(TRTMNT)
- Find and Open a Copy file
- Check variable references:
  - **Scroll to find some variable defined in a copybook**
    - **PATIENT-ID, VALID-BILLABLE-TYPES**
  - **Right-click over the variable name and select Open Declaration**



Use Case #2 – **Set up the default JCL Job Card for your LPAR**

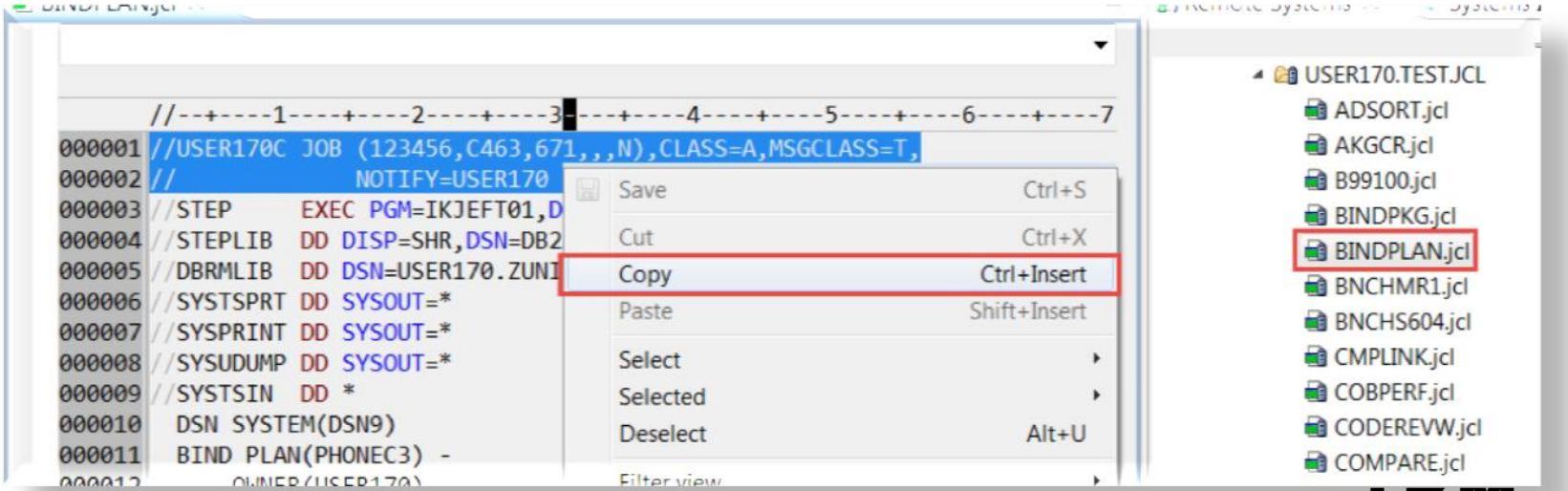
Remote z/OS File Search invokes SuperC in Batch (through JCL). So you need to provide a valid Job Card for your LPAR

Steps:

- **Expand MVS Files**
- Expand a JCL (source library) file
- Double-click and open the file

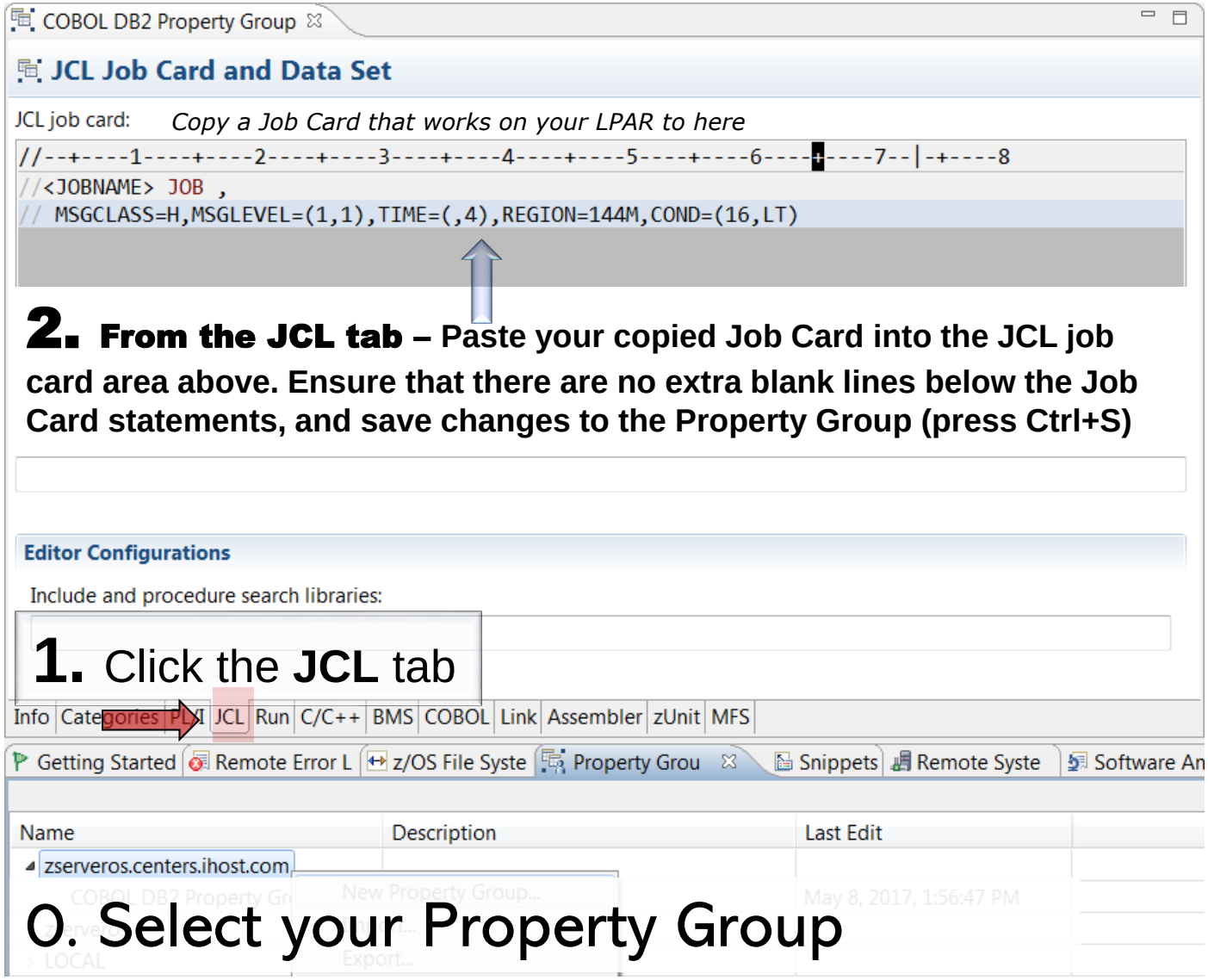
From within the open JCL file:

- Select your Job Card
  - All parts of it: //JOBLIB, etc.
- Right-click and copy the selected text into the Windows paste-buffer



Set up the default JCL Job Card for your LPAR – continued

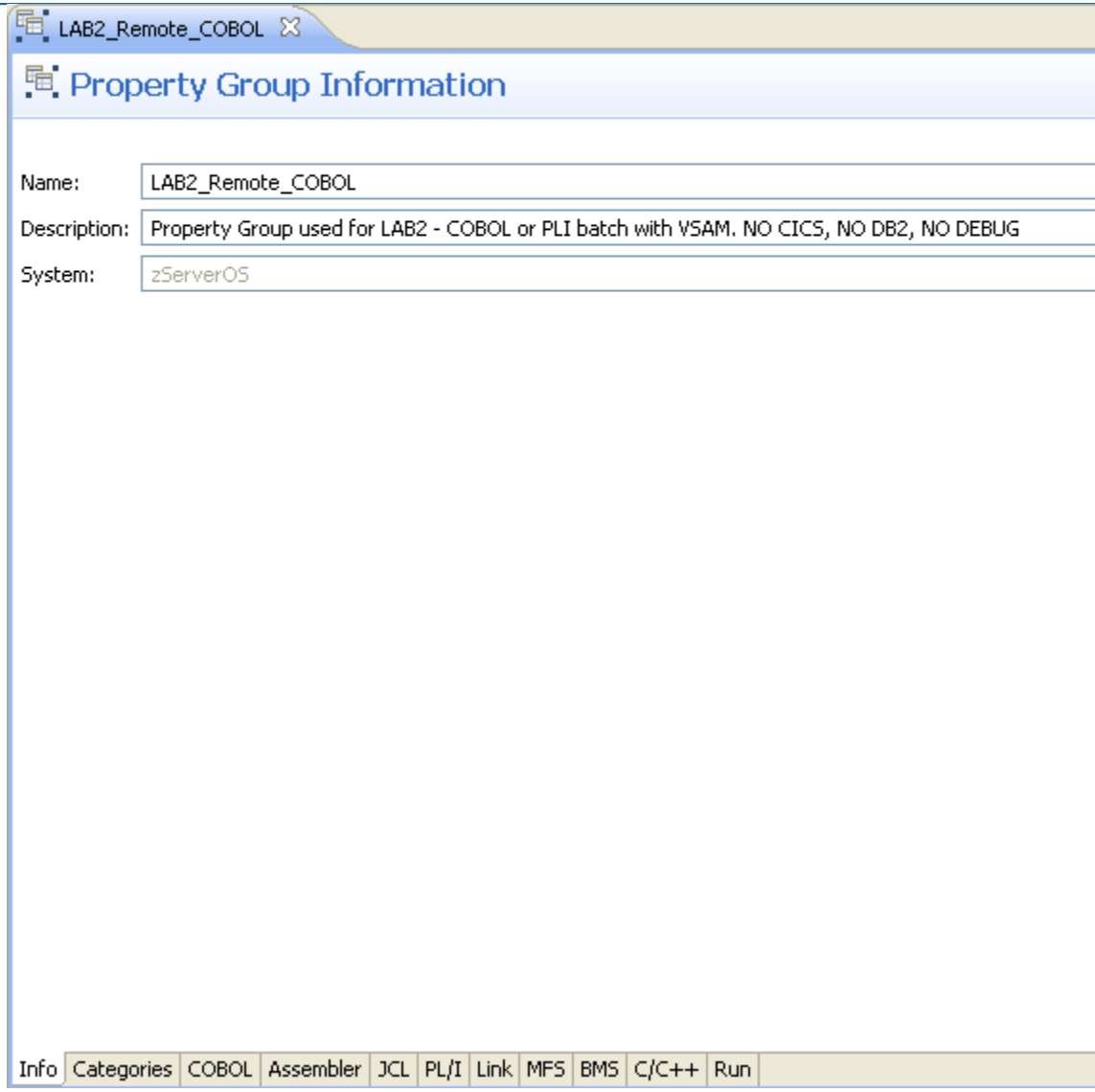
From the JCL tab within your Property Group



Use Case #3 – Customizing Property Groups for Build

- This dialog shows all of the possible languages, 3270 screen technologies and batch link and run-time options you might wish to customize during this editing session as tabs.

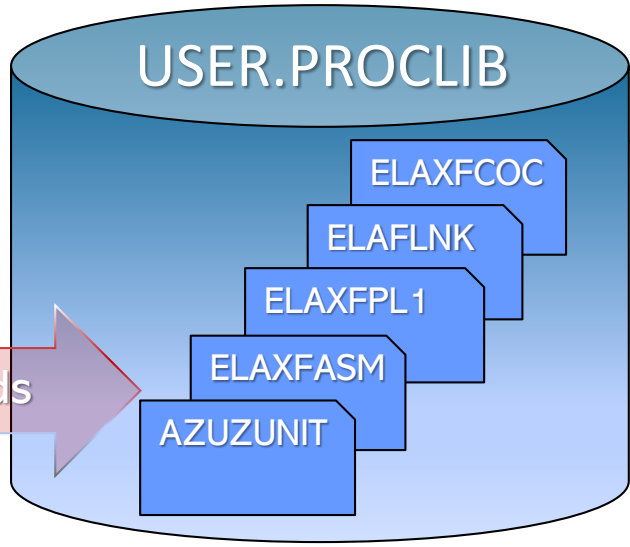
- Note that the Categories tab contains checkboxes you can de-select to remove configuration settings for technologies and run-times you don't need for your work
  - Languages
  - Products
  - The "GO" step in batch JCL
  - Etc.



Property Group Entries for Build – Customized properties, specific to your LPAR

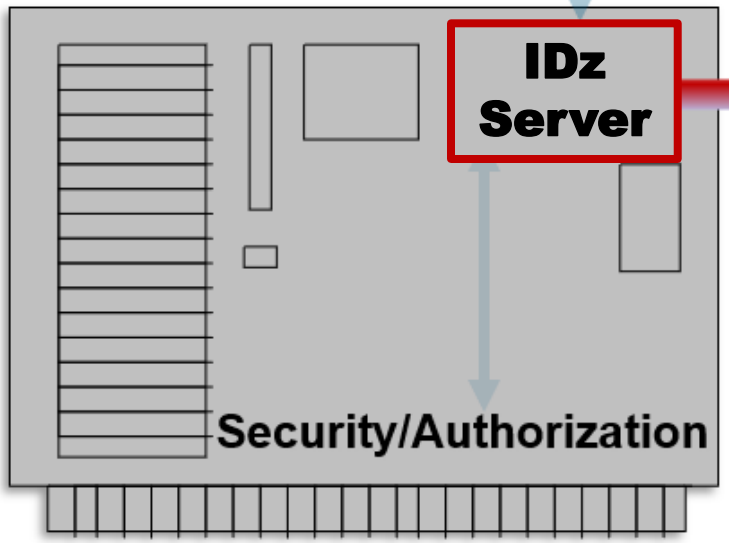
```
SYSLIB:  My-Copy-PDS Test-Copy-PDS-Test Prod-Copy-PDS DCLGEN-PDS
JCL:     Job Card, PROC-Search-PDS-1 PROC-Search-PDS-2
COBOL:   Compile-PARMS, SYSLIB, Called-Pgm-Lib-1 Called-Pgm-Lib-2
PLI:     Compile-PARMS, Includes, Called-Pgm-Lib-1 Called-Pgm-Lib-2
LINK:    PARMS, Link Libraries, Load Library
ZUNIT:   AZURES-PDS, AZUCFG-PDS, COBOL-TestCase-Gen-PDS
BMS:     DSCTLIB-PDS, Object-PDS, //SYSLIB
MFS:     FORMAT-PDS, Listing-PDS, //SYSLIB
...
```

Overrides for DD Cards



Installed IDz PROCs

IDz Client passes Property Group entries to the IDz server which File-Tailors the installed IDz PROCs



MVS Resources

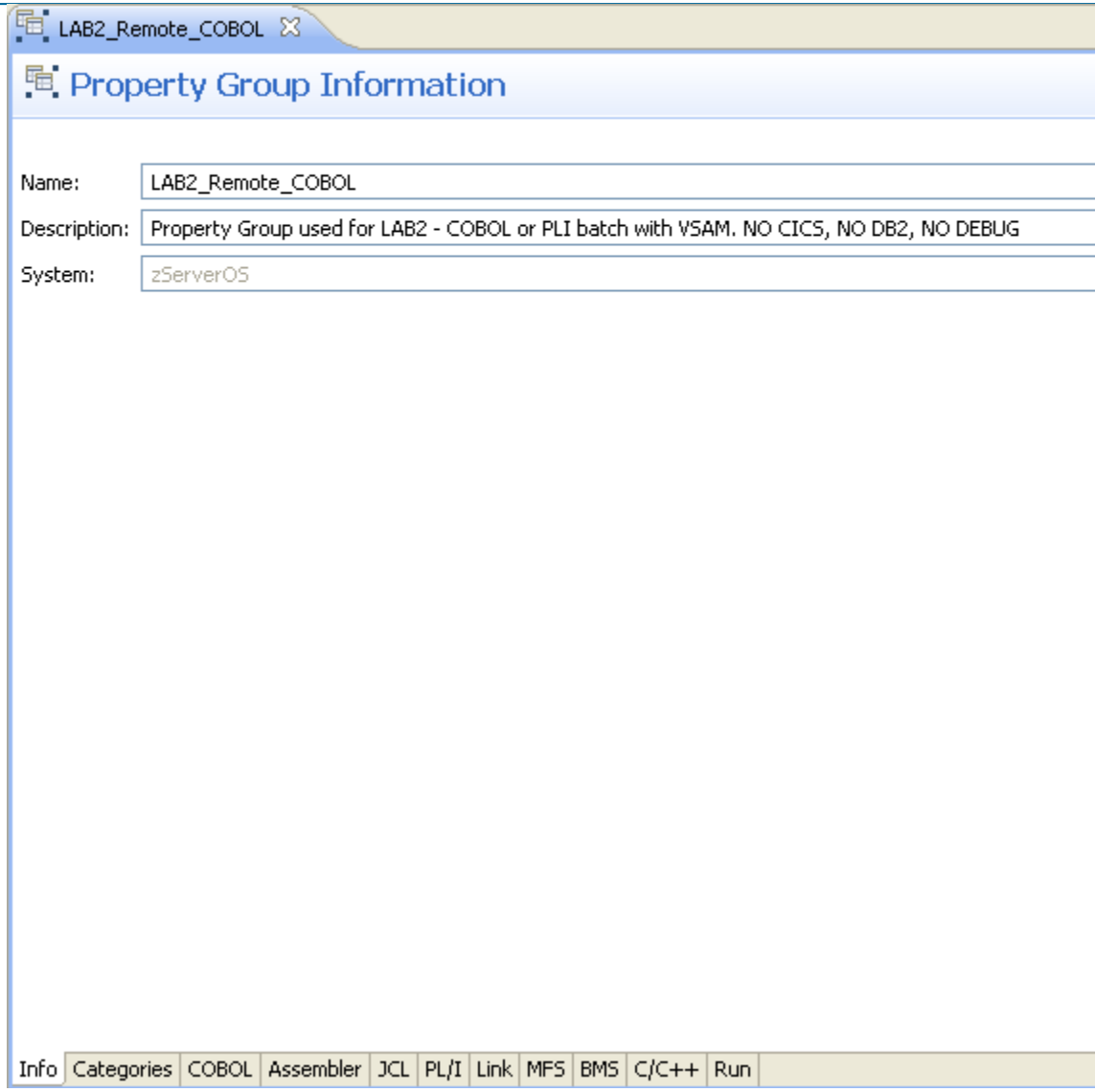
z/OS Datasets + access to:

- TSO
- Build (Compile/Link) Procs
- Endevor, SCLM, RTC EE
- Changeman, Panvalet, etc.
- MVS Utilities
- CLISTs/REXX Execs

Customizing Property Groups for Build

- This dialog shows all of the possible languages, 3270 screen technologies and batch link and run-time options you might wish to customize during this editing session as tabs.

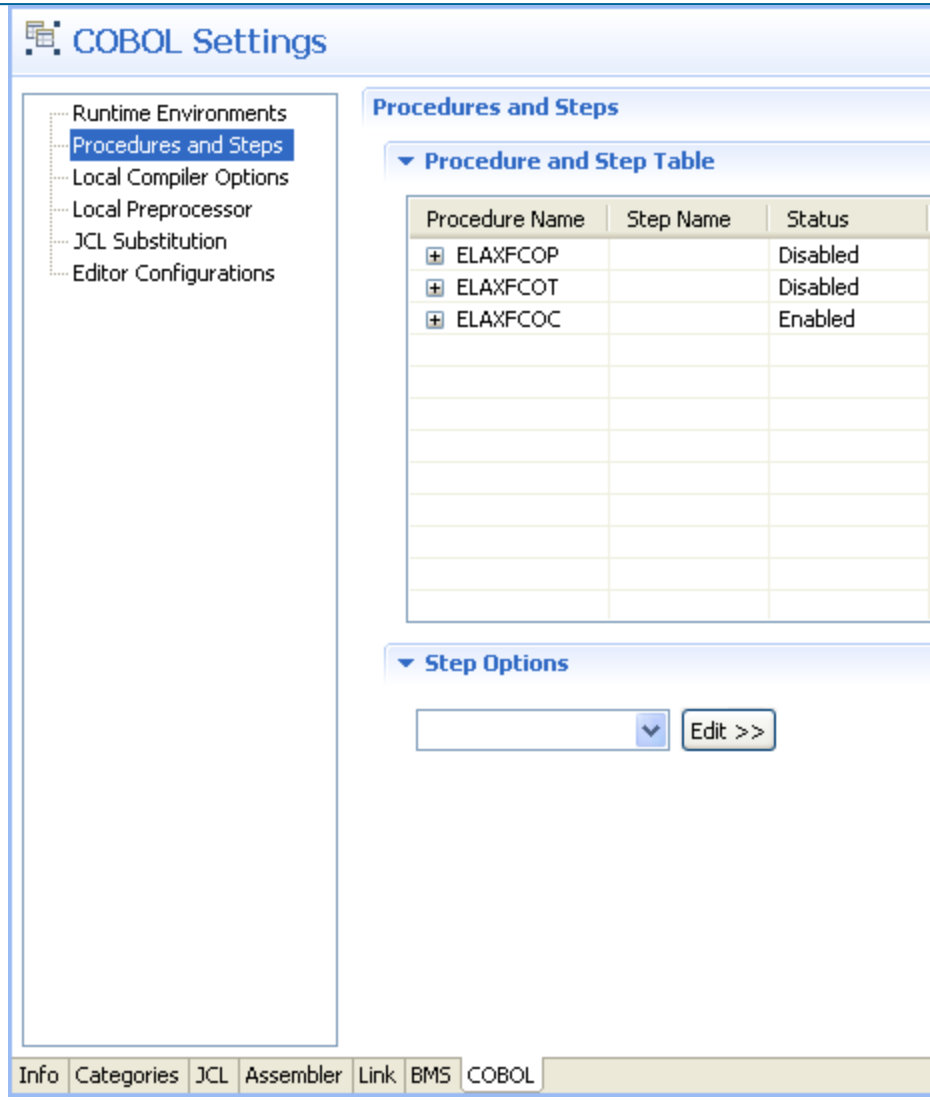
- Note that the Categories tab contains checkboxes you can de-select to remove configuration settings for technologies and run-times you don't need for your work
  - Languages
  - Products
  - The "GO" step in batch JCL
  - Etc.



Customizing Property Groups for Build

The COBOL Settings tab has several sub-tabs and options:

- **Runtime Environments:**
  - Check for each that apply:
    - **Ex. CICS and DB2, or IMS and DB2**
- **Procedures and Steps**
  - Allows you to customize your compile PROCs (details on the next slide)
- **Local Compiler Options**
  - Allows you to customize the compiler settings for local COBOL (Windows executable) applications
- **Local Preprocessor**
  - Use for custom code preprocessing such as:
    - **Substituting copy statements for ++INCLUDE for Local Syntax Check**
    - **Not used for EXEC CICS or EXEC SQL**
- **JCL Substitution**
  - Allows you to over-ride (add, edit and remove custom variables) from the default JCL generated through the wizards
  - These are accessed through a SET statement
- **Editor Configurations**
  - Provides a mechanism to setup and configure support for custom/macro pre-processing:
    - **Remote (REXX) pre-processor support**
    - **Local (C++ or Java) pre-processor support**
  - Specifying program library concatenation for opening Called programs

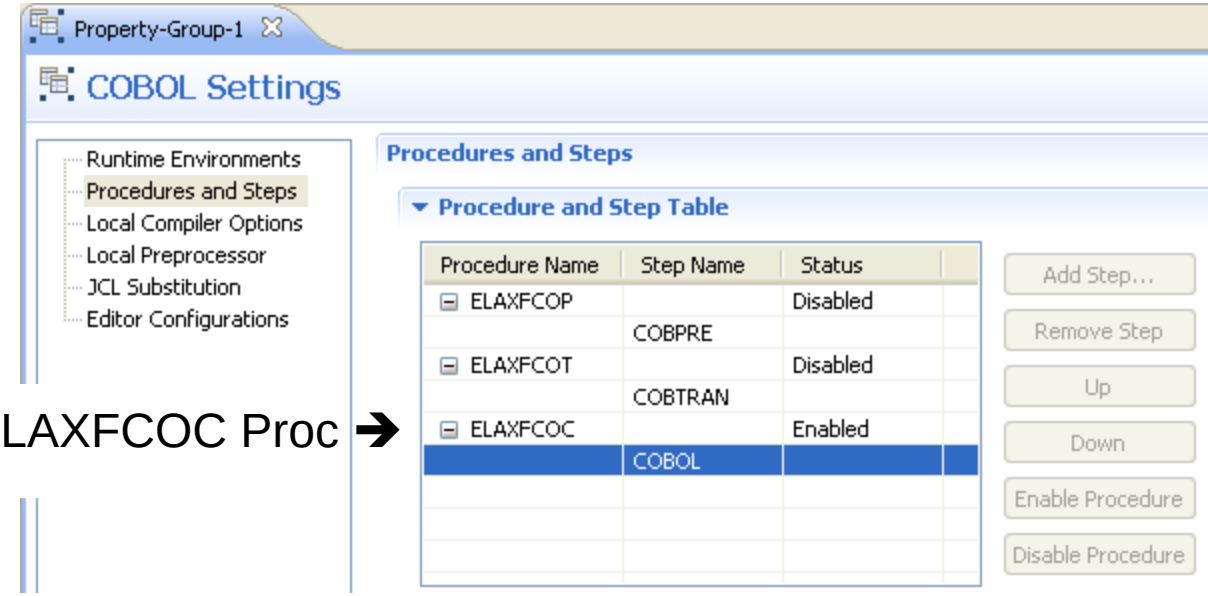


Customizing Property Groups for Build

- When you click the Procedures and Steps option you have access to three JCL Procs – that were installed and customized by your MVS Systems Programming staff when they installed and configured the IDz z/OS components.
  - **ELAXFCOP** – A Proc which invokes the DB2 Pre-processor
  - **ELAXFCOT** – A Proc which invokes the CICS Pre-processor
  - **ELAXFCOC** – A Proc which invokes a COBOL Compile/Link/Bind

    ELAXFCOC also will invoke:
    - **DB2 Co-processor** (used instead of the Pre-processor if the right levels of system software installed on your z/OS) – and if EXEC SQL statements are in your code
    - **CICS Translation**

You will typically only use the ELAXFCOC Proc ➔

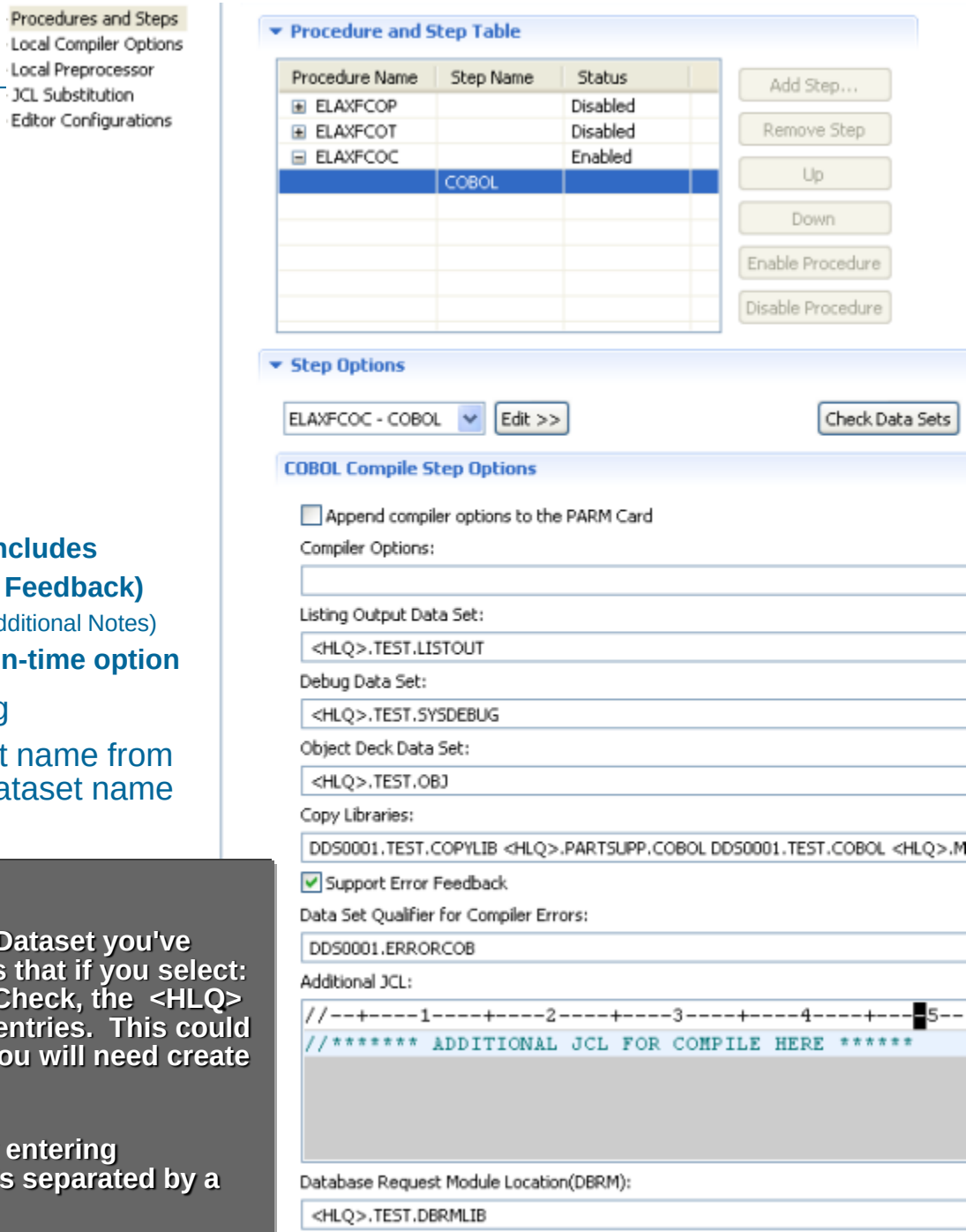


- **Info on the DB2 Co-processor**
  - http://pic.dhe.ibm.com/infocenter/comphelp/v111v131/index.jsp?topic=%2Fcom.ibm.aix.cbl.doc%2FPGandLR%2Fconcepts%2Fcpdb203.htm

Customizing Property Groups for Build

From the Procedure and Step Table, Open ELAXFCOC, select **COBOL** and customize:

- The compile Proc name
- The compile resolved-JCL Proc step name
  - **By default: COBOL**
- Compiler options
- Various compiler DD cards for:
  - **Listing dataset**
  - **The OBJ library PDS**
  - **//SYSLIB - The library for copybooks and includes**
  - **A sequential file for Compiler Errors (Error Feedback)**
    - **Best Practice: Hard code your TSO ID** (see Additional Notes)
  - **//DBRMLIB – if you've selected DB2 as a run-time option**
- Click: **Check Data Sets** – to verify spelling
- **Even better –** you can drag & drop a dataset name from the Remote Systems view, to populate the dataset name fields **(next slide)**



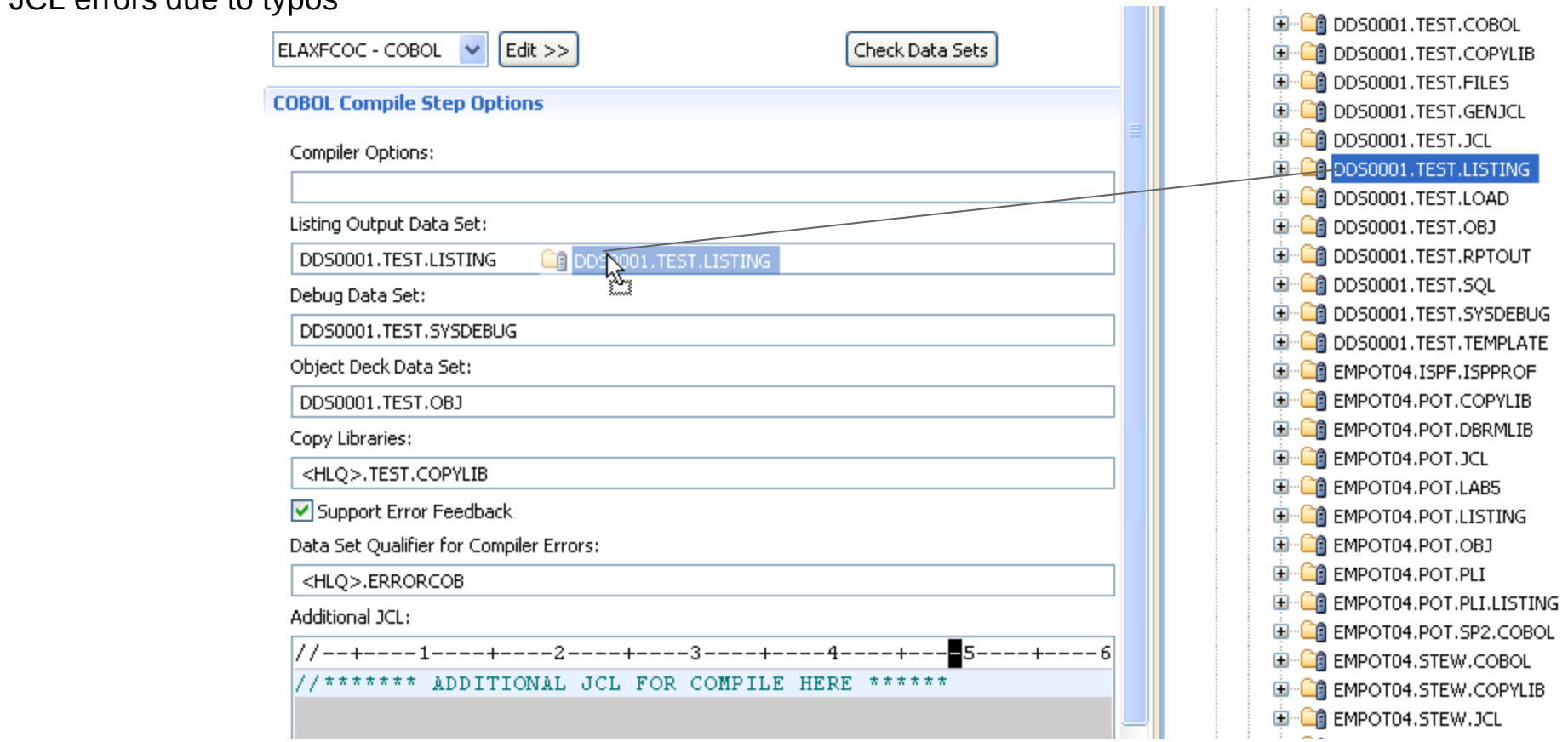
Additional Notes:

<HLQ> will resolve to the High Level Qualifier of the Dataset you've selected from Remote Systems Explorer. This means that if you select: MYCORP.TEST.COBOL(DTEVAL) for Remote Syntax Check, the <HLQ> variable resolves to MYCORP for all Property Group entries. This could be significant for the Compiler Errors Data Set – as you will need create authority against MYCORP as a high-level qualifier.

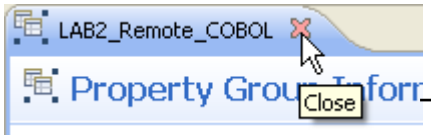
You can concatenate datasets in Copy Libraries by entering additional dataset names to the right of existing DSNs separated by a space (blank)

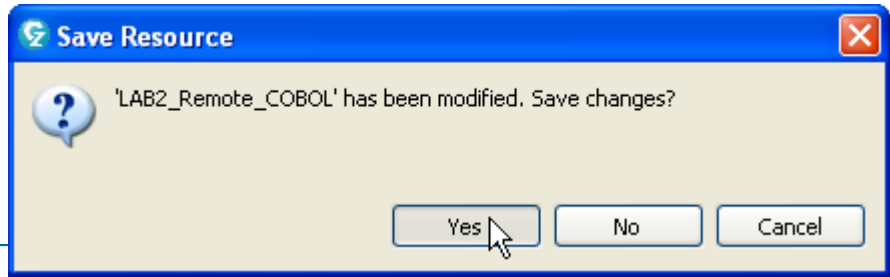
Customizing Property Groups for Build

- If the datasets exist for your compile outputs, you can just select, left-click hold, drag and drop them from the Remote System view into the appropriate Data Set name fields. This will be more productive and help avoid JCL errors due to typos



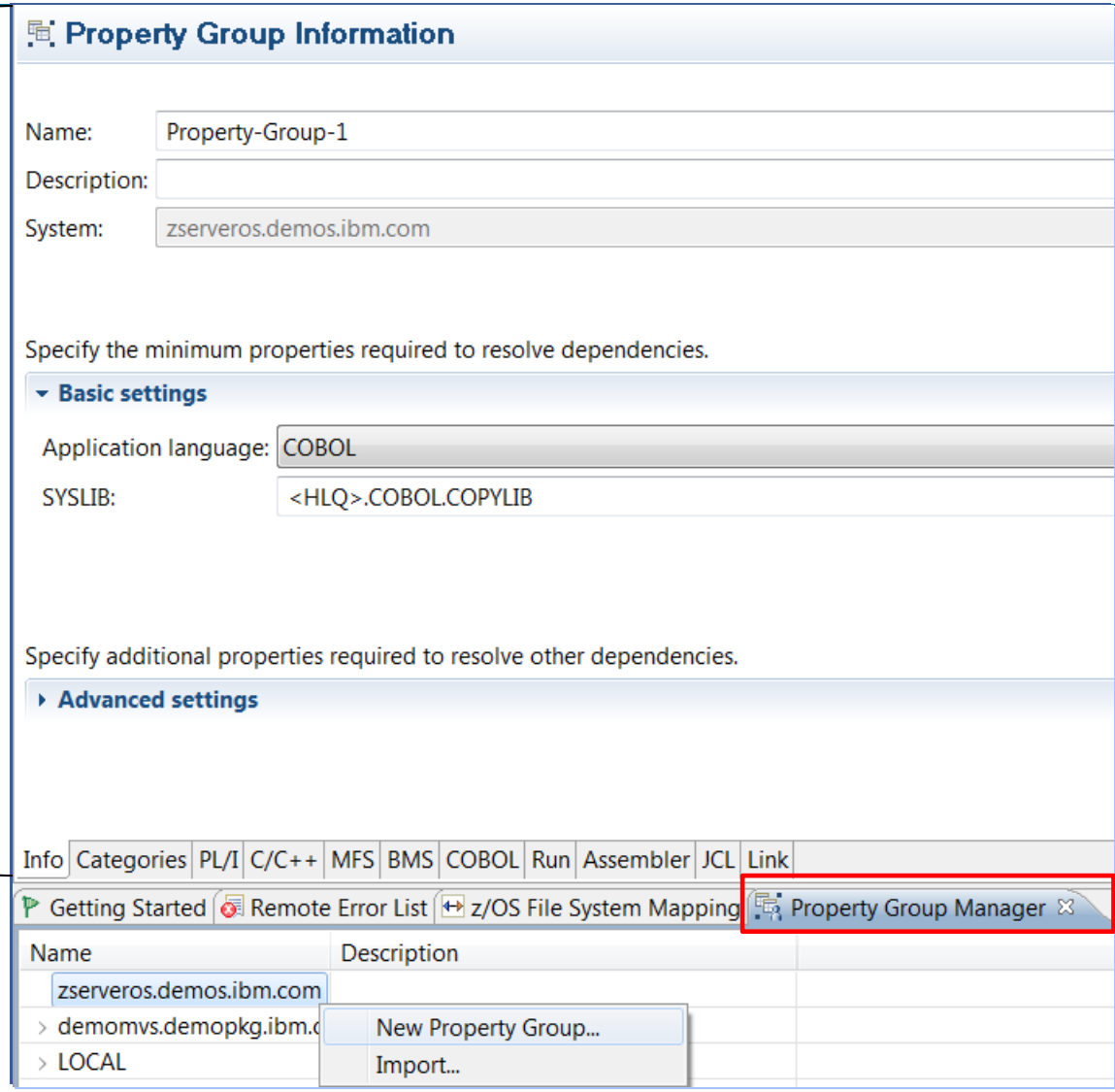
- When you are finished working, close the Edit Area, and **save changes**





Steps - 1. Create a New Property Group

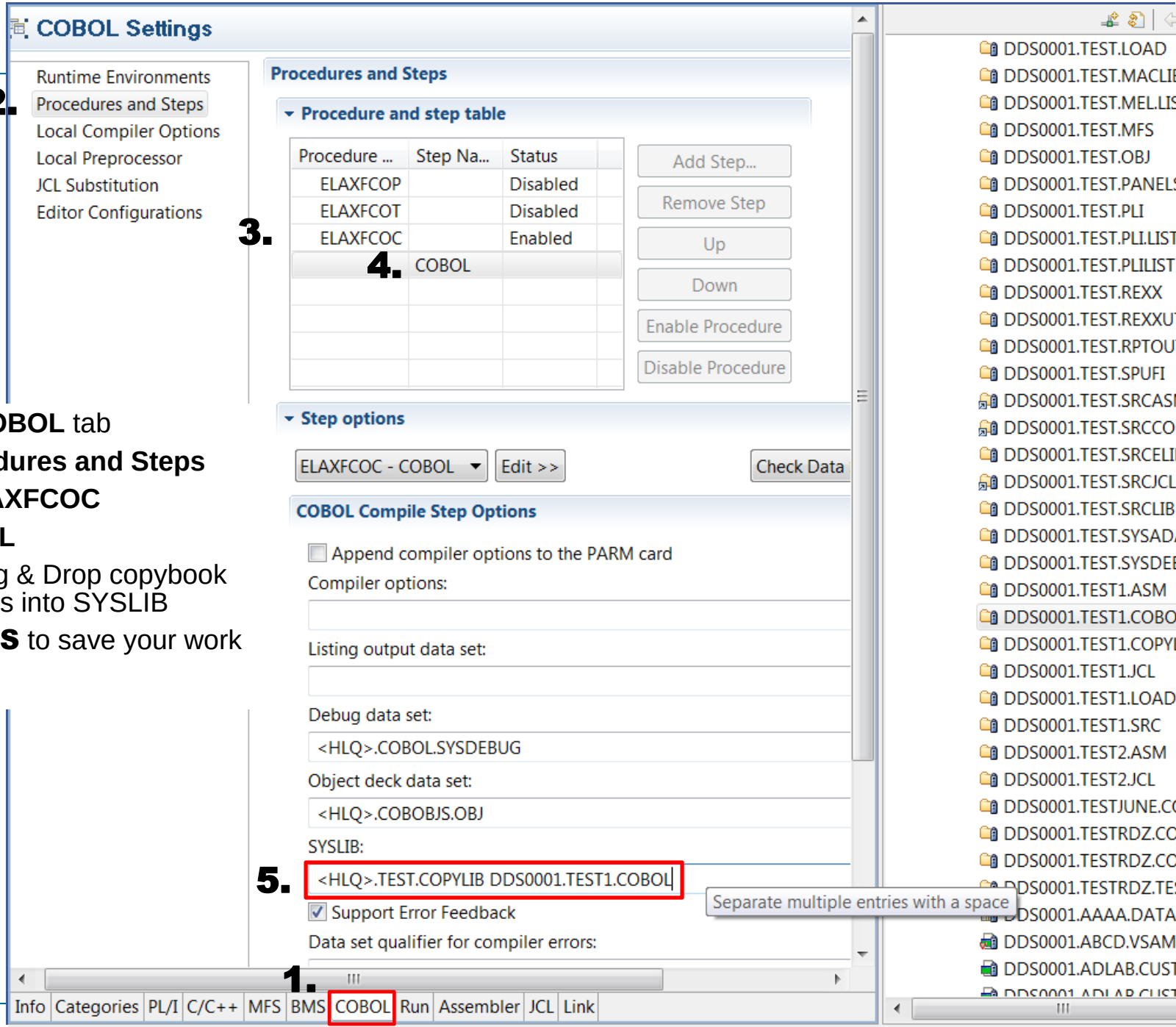
A new Property Group for a connection



From Property Group Manager:

- Right-click over your connection
- Select New Property Group…

2. Modify SYSLIB



1. From the **COBOL** tab
2. Click **Procedures and Steps**
3. Expand **ELAXFCOC**
4. Click **COBOL**
5. Type or Drag & Drop copybook library names into SYSLIB
6. Press **Ctrl+S** to save your work

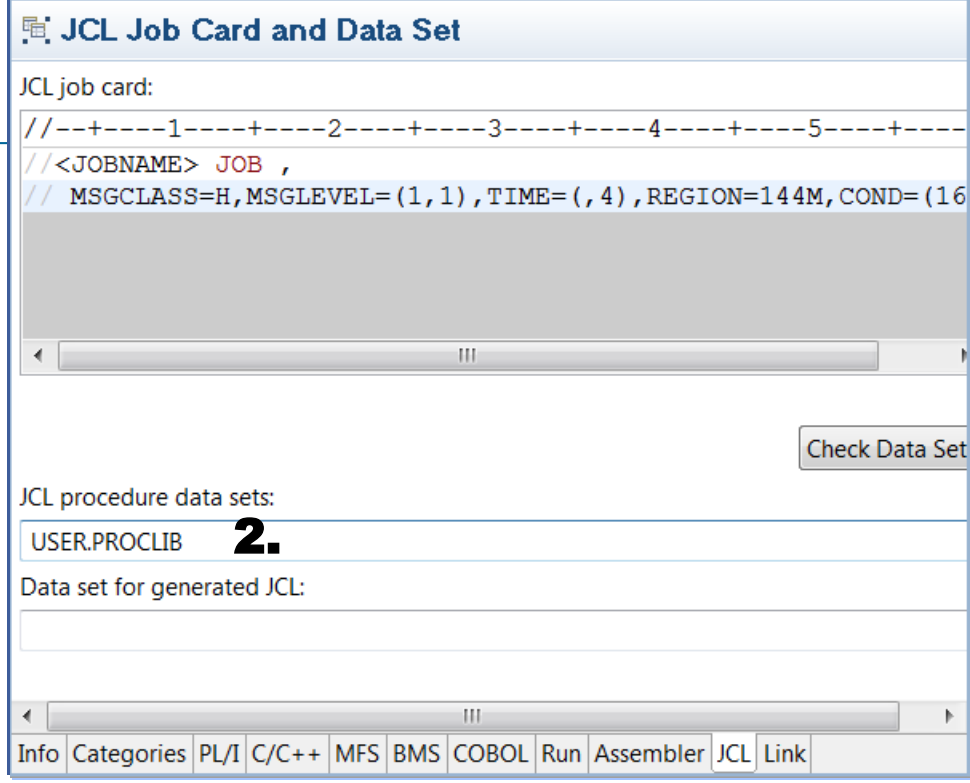
3. Add PROC Library Lookup Paths

Let's get the Open PROC function to work…

From your Property Groups:

1. **From the JCL tab**

2. **From JCL procedure data sets;** enter the DSN of one or more JCL PROCLIBs – each separated by a space (if entering multiple)



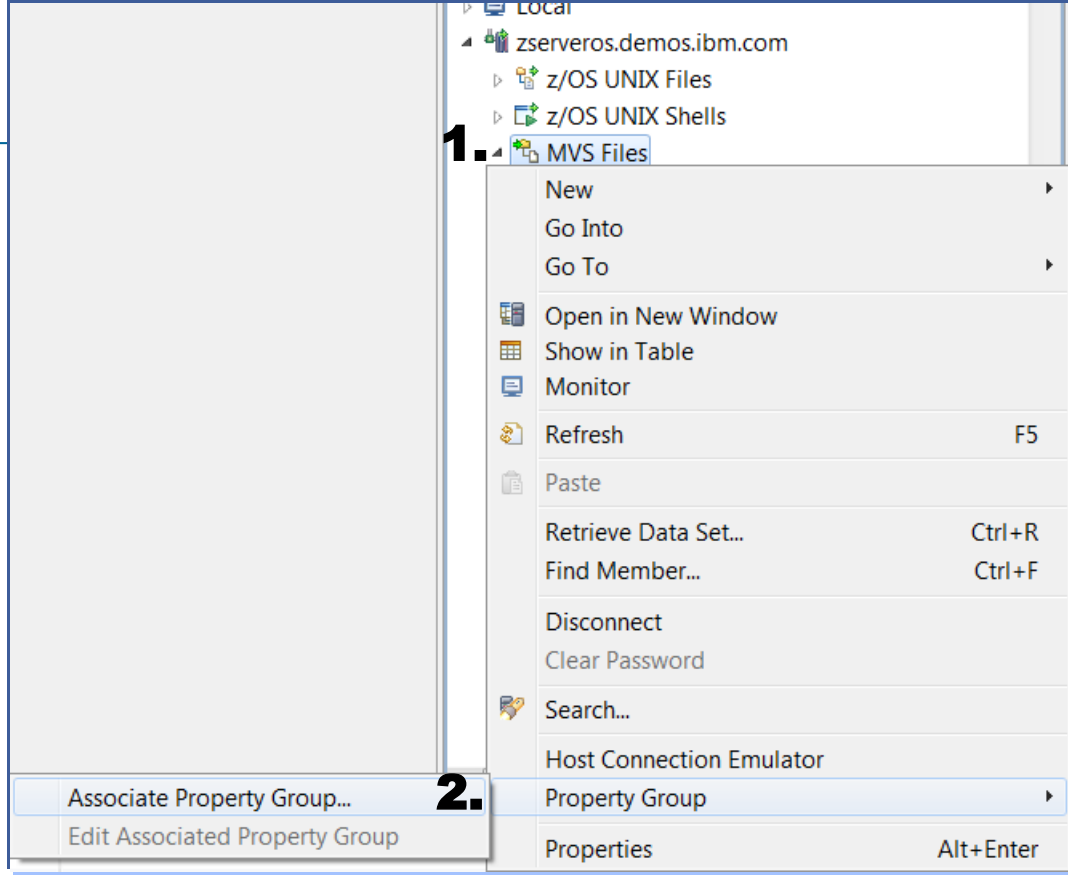
- **T**est your work by opening a run-stream JCL file that has PROC statements:
  - Find a PROC statement
  - Select the called **PROC name**
  - Right-click and select Open JCL Procedure
  - See next slide for an example…

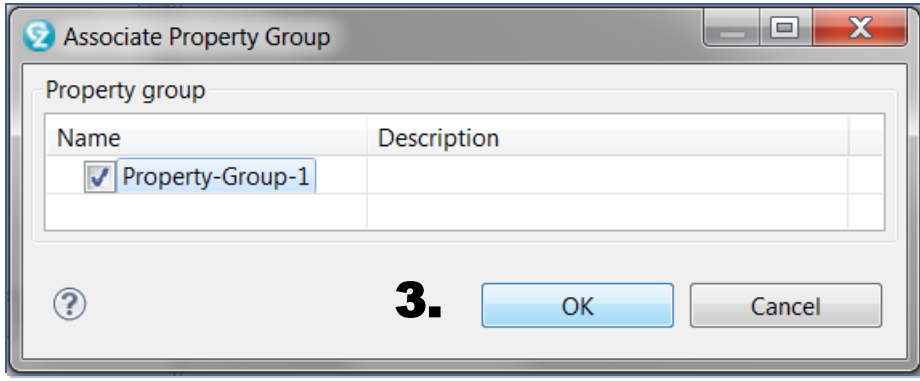
4. Assign Property Group to MVS Files

Assign the Property Group to the z/OS Resources you want it to manage

Steps:

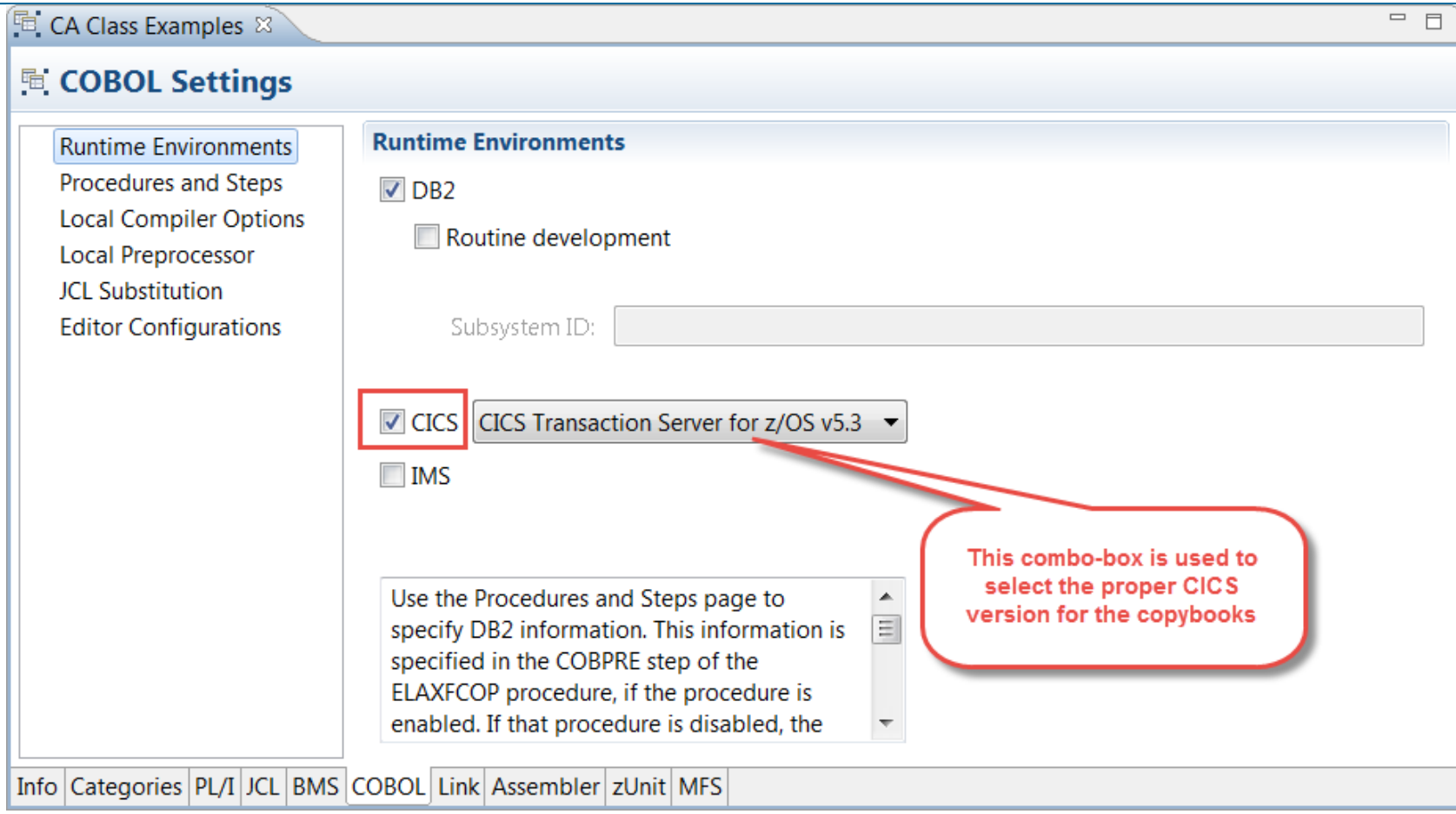
- From Remote Systems view
  1. **Right-click over MVS Files**
  2. Select **Property Group >**
     - **Associate Property Group…**
  3. **Check the Property Group** you want assigned and click **OK**





☑ When you're finished defining your Property Group and assigning it, test your work by opening a program that has Copy/Include statements, to ensure that the Property Group's SYSLIB concatenation identifies all of the libraries your references to Copy members needs.

Property Groups and CICS System Copy Files – **DFHEIBLK, DFHCOMMAREA**



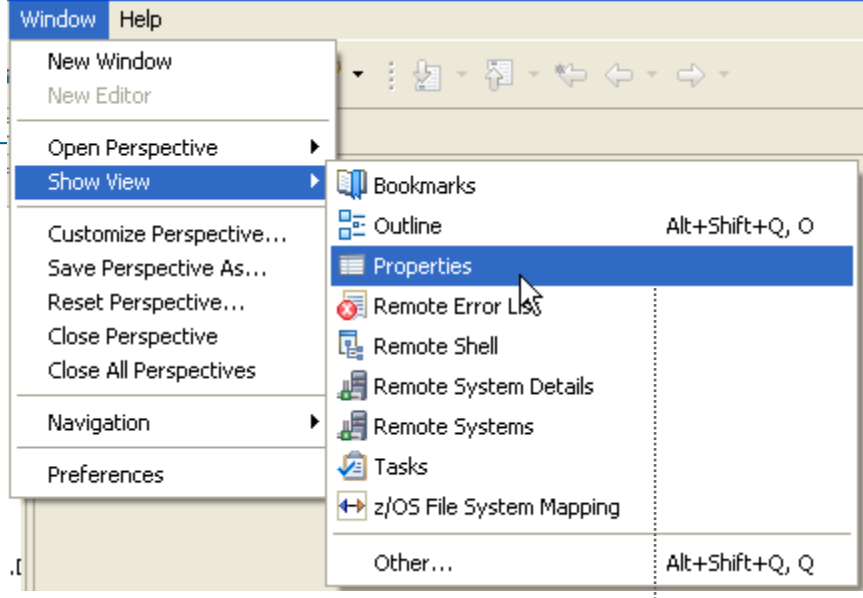
- If your shop depends on the CICS pre-compiler to supply system copybooks you will need to:
  - Check the CICS box, in the COBOL Property Group tab
  - Select your CICS TS version – in order that IDz unpacks the correct release of the copybooks for your application code

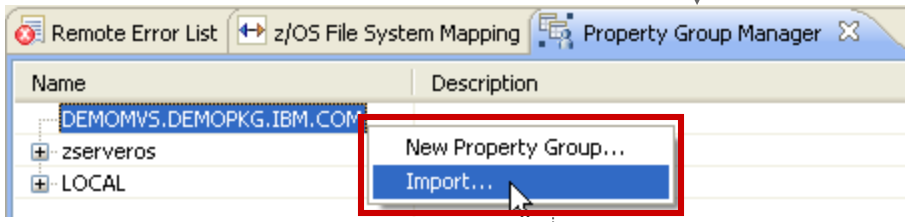
Another Option - **Import a Property Group**

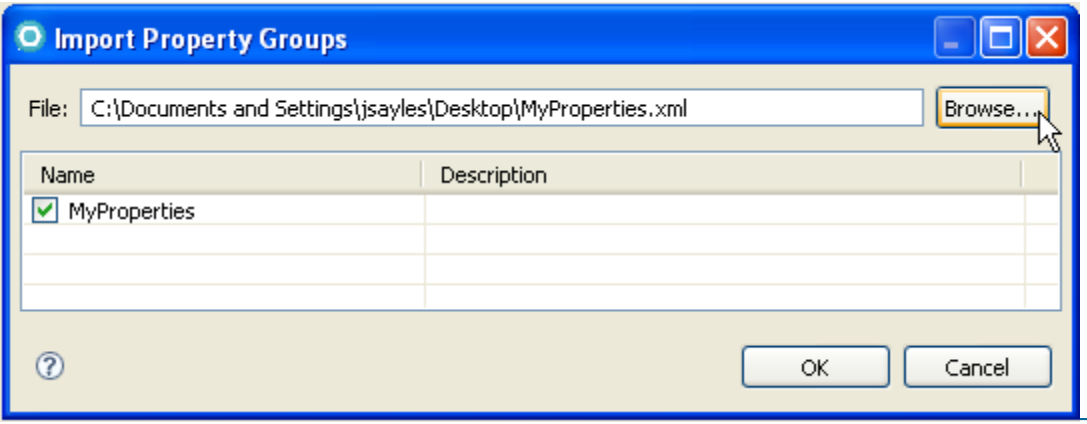
- If you are using IDz at a company, it is likely that someone has already set up your Property Groups
- However you still may be called upon to customize some of the entries, so let's find out how to import a property group and see what the settings are all about
  - From the Window menu, select:

    Show View >

    Properties
- This opens the Property Group Manager view which lists your connections.
  - Right-click over the connection you wish to create properties for, and select **Import…**
  - Click **Browse**, and select the **property.xml** file you wish to use







Note: For this class it is easier to create New Property Group files than to Import. But if your company has a specific set of Property Groups you should import them – as this could save a lot of time.