

z/OSMF Hands-On Labs

- Choose Your Own Topic -

Create your own Workflow (Workflow Editor)

Copyright© by SHARE Association Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license. <http://creativecommons.org/licenses/by-nc-nd/3.0/>

CC BY-NC-ND 1

Abstract:

Do you want to create your own z/OSMF Workflow? This lab will allow you to create your own z/OSMF Workflow using the z/OSMF Workflow Editor. The self-directed lab will show you how to add your own steps, variables, and jobs, and end up with your own Workflow that you can share with users!

The z/OSMF Workflow Editor is provided in z/OS V2.2 PTF UI42847 (closed January 3, 2017), and the z/OSMF V2.1 PTF UI43814 (closed February 1, 2017).

Some basic terms to get you started:

- **Workflow:**
 1. An activity associated with the z/OS system, such as configuring a component or product.
 2. The instantiation of a workflow in z/OSMF, based on a workflow definition. A workflow consists of one or more units of work to be performed on the z/OS system, as described by the workflow definition. A workflow is created when the Workflows task in z/OSMF is used to create an instance of a workflow from a supplied workflow definition file.
- **Workflow definition:** The logical structure of a workflow, represented through a series of steps, through the main XML file (the workflow definition file). The workflow definition identifies the various system objects and actions that constitute activities on z/OS and the rules for performing the activities.
- **Workflow variable input file:** An optional file that supplies default values for one or more of the input variables defined in the workflow definition file.
- **Step:** A single, logical unit of work in a workflow.

Exercise instructions

When you follow this self-directed lab, here is a high-level overview of what you will learn:

- ___ 1. With the z/OSMF Workflow Editor, create a Workflow from scratch.
- ___ 2. Import a step from the Step Library.
- ___ 3. Create steps in your Workflow.
- ___ 4. Define variables to be used in your Workflow.
- ___ 5. Provide defaults and variable specifications.
- ___ 6. Save your steps into a library for others to use.
- ___ 7. Retrieve steps from a library that others have provided.
- ___ 8. Save your Workflow.
- ___ 9. Try your Workflow out to make sure it is as you desire.
- ___ 10. Edit an existing Workflow to add, remove, and change steps.
- ___ 11. Try your Workflow out to make sure the changes are desirable.

Sample Workflow for this lab:

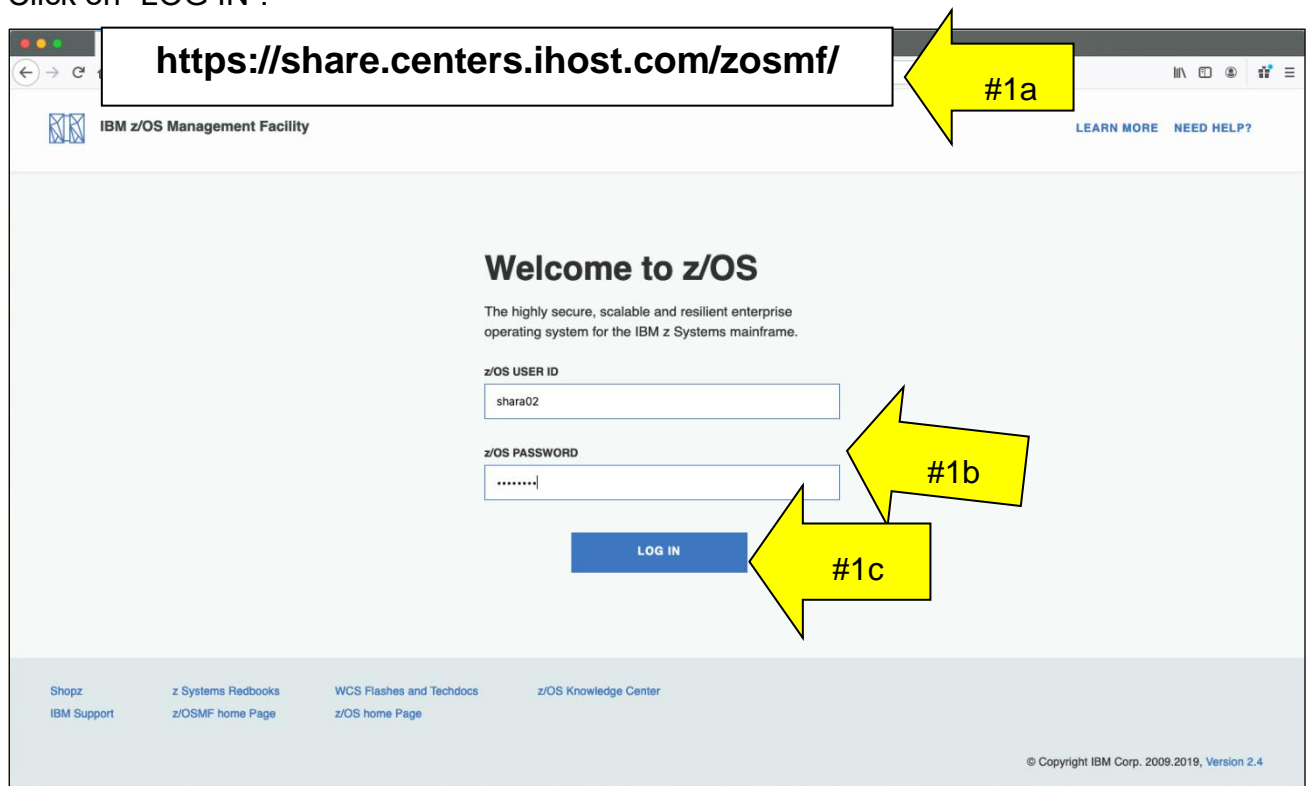
The Workflow that we will create in this lab is very simple. It will:

- 1. Create and run a step from the Step Library. (Using a REST API)
- 2. Run LISTCAT against a dataset. (Using JCL)

1. Logon to z/OSMF.

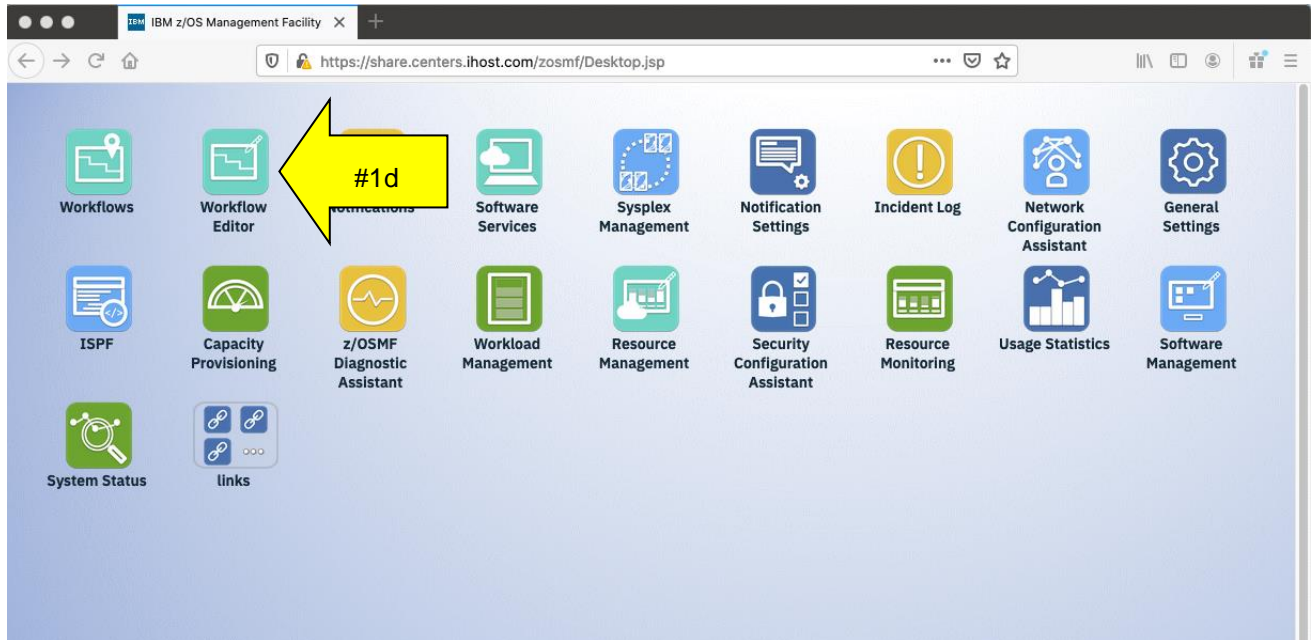
In this step, we will now go into z/OSMF to use the Workflow Editor function. For this lab, we are using a z/OSMF V3.1 system.

- a. Go to <https://share.centers.ihost.com/zosmf/> on the Firefox or IE web browser. (If you want to follow this lab on your own system, that is fine. Just note some of the samples we use you will need to supply yourself, using the Appendix to find those samples.).
- b. Using the userid you were given (SHARAnn, SHARBnn, or SHARCnn) and the password, logon to z/OSMF. The userid you were given is a regular z/OS userid on this system and has been given access to z/OSMF. There is *no* z/OSMF code on this workstation, all executables (except the web browser) is on the z/OS system.
- c. Click on “LOG IN”.



Enter into the Workflow Editor function.

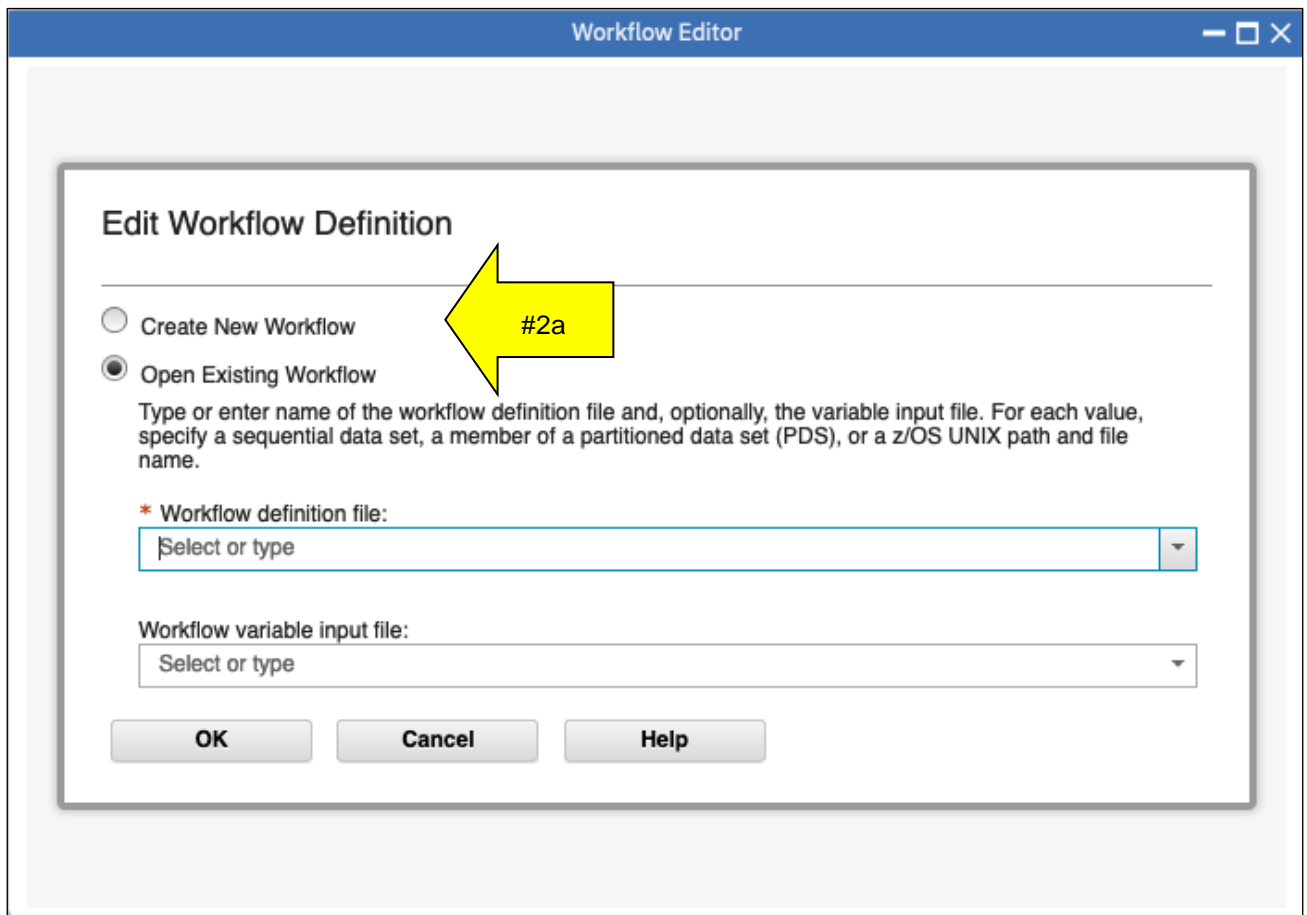
d. Click on “Workflow Editor”.



2. Start to create your new Workflow.

In this step, we will create a brand-new Workflow. You can see from this screen that you also could select an existing Workflow to edit. For now, we will create a Workflow from scratch.

- a. Click on “Create New Workflow”. By default, you can see that “Open Existing Workflow” is selected. We’ll use that option later.

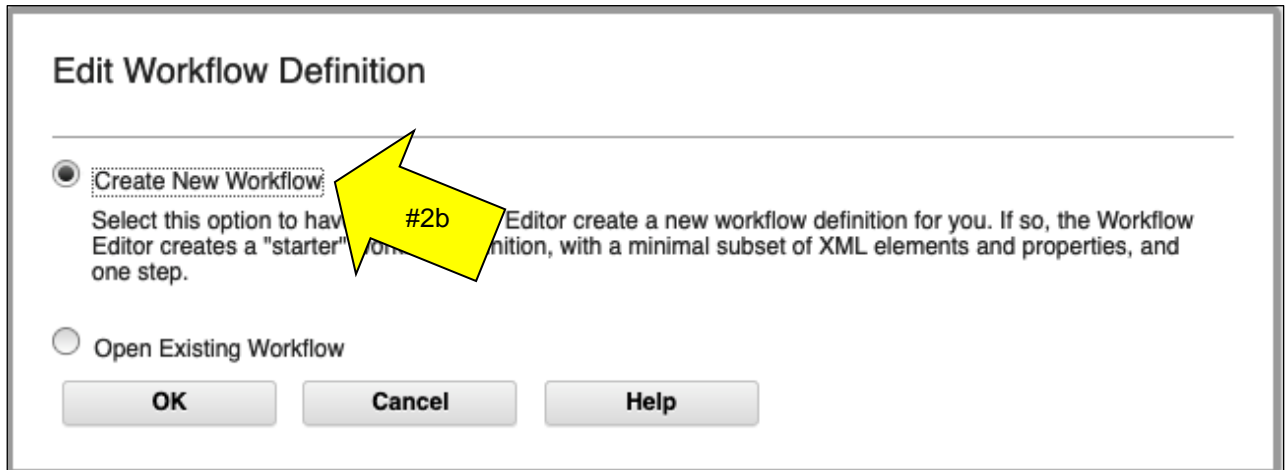


The screenshot shows a window titled "Workflow Editor" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window is a dialog box titled "Edit Workflow Definition".

Inside the dialog box:

- There are two radio buttons:
 - ☐ Create New Workflow
 - ☒ Open Existing Workflow
- A yellow arrow points from the right towards the "Create New Workflow" radio button. The arrow contains the text "#2a".
- Below the radio buttons is a text instruction: "Type or enter name of the workflow definition file and, optionally, the variable input file. For each value, specify a sequential data set, a member of a partitioned data set (PDS), or a z/OS UNIX path and file name."
- Below the instruction is a label: "* Workflow definition file:"
- Below the label is a text input field with the placeholder text "Select or type" and a dropdown arrow on the right.
- Below that is a label: "Workflow variable input file:"
- Below the label is another text input field with the placeholder text "Select or type" and a dropdown arrow on the right.
- At the bottom of the dialog box are three buttons: "OK", "Cancel", and "Help".

-
- b. When you “Create New Workflow” this is the next dialog box that appears. Click on “OK”.



Metadata for your Workflow.

At this point, we need to provide some general information about our new Workflow, called “Metadata”, as indicated on the tab. There are other tabs for “Steps”, “Variables”, and “Input Properties”. You can see that there is even a scroll bar on the right, as we can’t even see all the information that we could provide here. The * fields are required. Make sure you scroll down to see more options.

At any time, you can click on the upper right “Help” to read more about each of these fields.

- c. Workflow ID: type a meaningful name for this required field.
 - A suggested name: **YourUserID Workflow Editor Lab**, where **YourUserID** is your assigned userid, such as SHARA02.
- d. Let’s make this Category **General**. A General Category just means it’s not identified as doing Configuration or Provisioning. General is fine for your first Workflow, and what you probably want to use for your own Workflows that perform normal tasks on your systems.
- e. Scope will be **System** here. System means that a maximum of one instance of this workflow can exist on any one system in the sysplex.
- f. Type in something you like for **Description**. If you type too much, the Editor will tell you it’s invalid.
- g. Scroll down to see more information.

The screenshot shows the 'Workflow Editor' window with the 'Metadata' tab selected. The window title is 'Workflow Editor'. In the top right corner, there is a 'Help' button and a pink arrow pointing to it labeled 'Help!'. The 'Workflow Definition:' section contains several tabs: 'Metadata', 'Steps', 'Variables', 'Feedback', and 'Input Properties'. The 'Metadata' tab is active, showing a text area with instructions: 'In a workflow definition, the metadata is used to identify the workflow and provide more information about it. On this tab, you can modify the workflow identifier, description, category, version, and other basic information for the selected workflow definition.' Below this, there are several fields and checkboxes:

- * Workflow ID:** A text field containing 'SHARA02 Workflow Editor'. A yellow arrow labeled '#2c' points to this field.
- Category:** A dropdown menu showing 'General'. A yellow arrow labeled '#2d' points to this dropdown.
- Scope:** A dropdown menu showing 'System'. A yellow arrow labeled '#2e' points to this dropdown.
- ☐ Callable Workflow
- Job Output Directory:** An empty text field.
- Range:** A dropdown menu showing 'System'.
- Default Name:** An empty text field.
- * Description:** A text area containing 'Now using the z/OSMF Workflow Editor.'. A yellow arrow labeled '#2f' points to this text area.
- Three checkboxes: ☐ Omit the prefix from instance variable references., ☐ Users can take ownership of steps automatically., and ☐ Contains parallel steps.

At the bottom of the window, there are four buttons: 'Save', 'Save As...', 'Test', and 'Cancel'. A yellow arrow labeled '#2g' points to the right side of the window, indicating the scroll bar.

- Do not click on “Save” at this time. This lab will take you through each tab.**

© Copyright IBM Corp. 2024

3. Adding Steps to your Workflow.

Now, we'll add some simple steps to our new Workflow.

- Click on the **Steps** tab. It might not be obvious now, but these tabs play a large role in creating your Workflow. You need to go through those tabs.
- You'll see that you already have a **"Starter-Step"** in this empty Workflow, given to you by default. That is because you have to have at least one step in a Workflow. Click on the **"Starter-Step"** and notice that **"Step Details"** appear at the right.
 - Notice that there are two detail "tabs" on the right: **Step Details** and **Variable Details**. Those two tabs on the right are very helpful to tell you what that step on the left is doing and what it concerns.

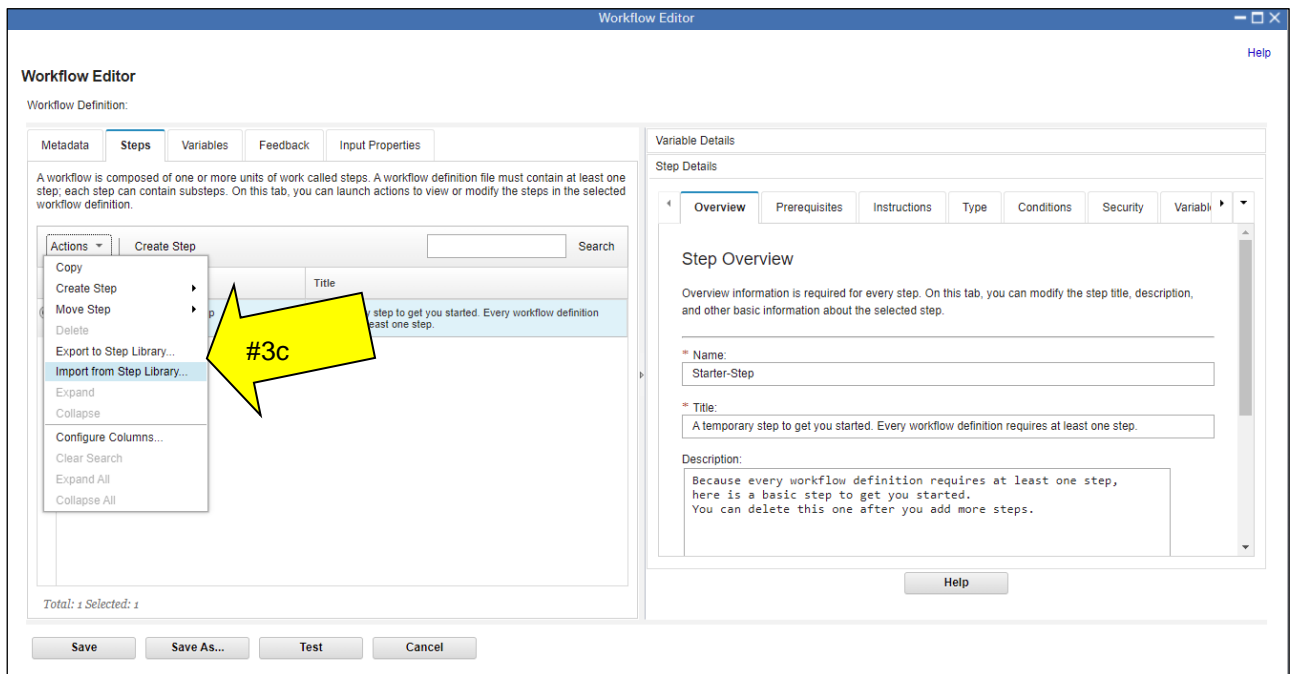
The screenshot displays the Workflow Editor interface. On the left, the 'Steps' tab is selected, showing a table with one step: '1 Starter-Step'. A yellow arrow labeled '#3a' points to the 'Steps' tab, and another yellow arrow labeled '#3b' points to the 'Starter-Step' row. On the right, the 'Step Details' panel is visible, showing the 'Overview' tab. A pink arrow labeled 'Look' points to the 'Step Details' panel. The 'Overview' tab contains fields for Name, Title, and Description. The Name field is 'Starter-Step', the Title field is 'A temporary step to get you started. Every workflow definition requires at least one step.', and the Description field contains a paragraph of text. At the bottom of the interface, there are buttons for 'Save', 'Save As...', 'Test', and 'Cancel'.

Step No.	Name	Title
1	Starter-Step	A temporary step to get you started. Every workflow definition requires at least one step.

Total: 1 Selected: 1

Save Save As... Test Cancel

- c. Click on **Actions->Import from Step Library...** to add the first “real” step to the new Workflow.



- d. A Step Library selection box will now appear with various tasks you can choose from.
- e. For this lab, Select IBM-Toolbox-ObtainZosRelease.
- f. Click Next

Import from Step Library

Find

No filter applied

	Name Filter	Description Filter	Creator Filter
<input type="radio"/>	IBM-Toolbox-CreateDataset	A step that allocates (creates) a partitioned data set (PDS) by submitting a batch job to run on z/OS.	IBM
<input type="radio"/>	IBM-Toolbox-CopyData	A step that copies data into a PDS member by using the IEBGENER utility.	IBM
<input type="radio"/>	IBM-Toolbox-CopyMember	This step copies a member from one data set to another by using the IEBRECFM utility.	IBM
<input checked="" type="radio"/>	IBM-Toolbox-ObtainZosRelease	This step obtains the z/OS release level for the local system by submitting a GET request for the z/OS version.	IBM
<input type="radio"/>	IBM-Toolbox-Shell	This step shows how to run a simple UNIX shell script.	IBM

Total: 5 Selected: 1

< Back Next > Cancel Manage Help

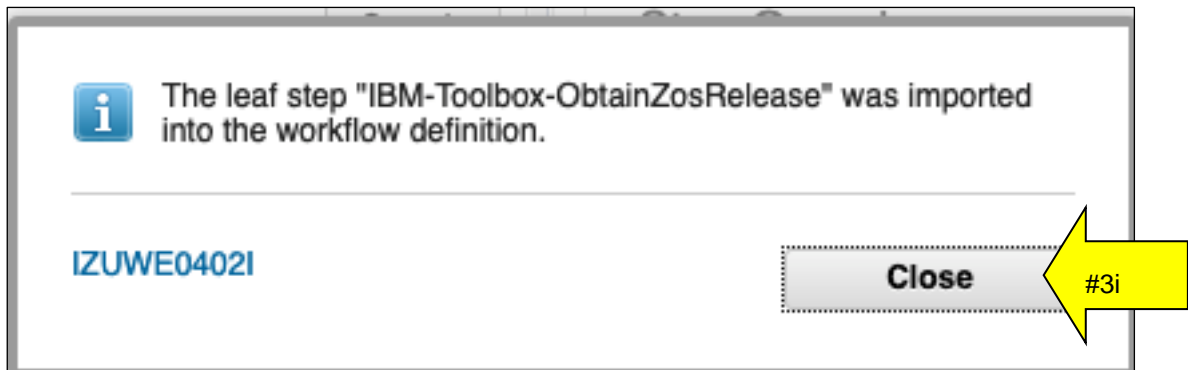
- g. Give your step a name, or you can use the default that is there.
- h. Click Finish

Import from Step Library

* Name:
IBM-Toolbox-ObtainZosRelease

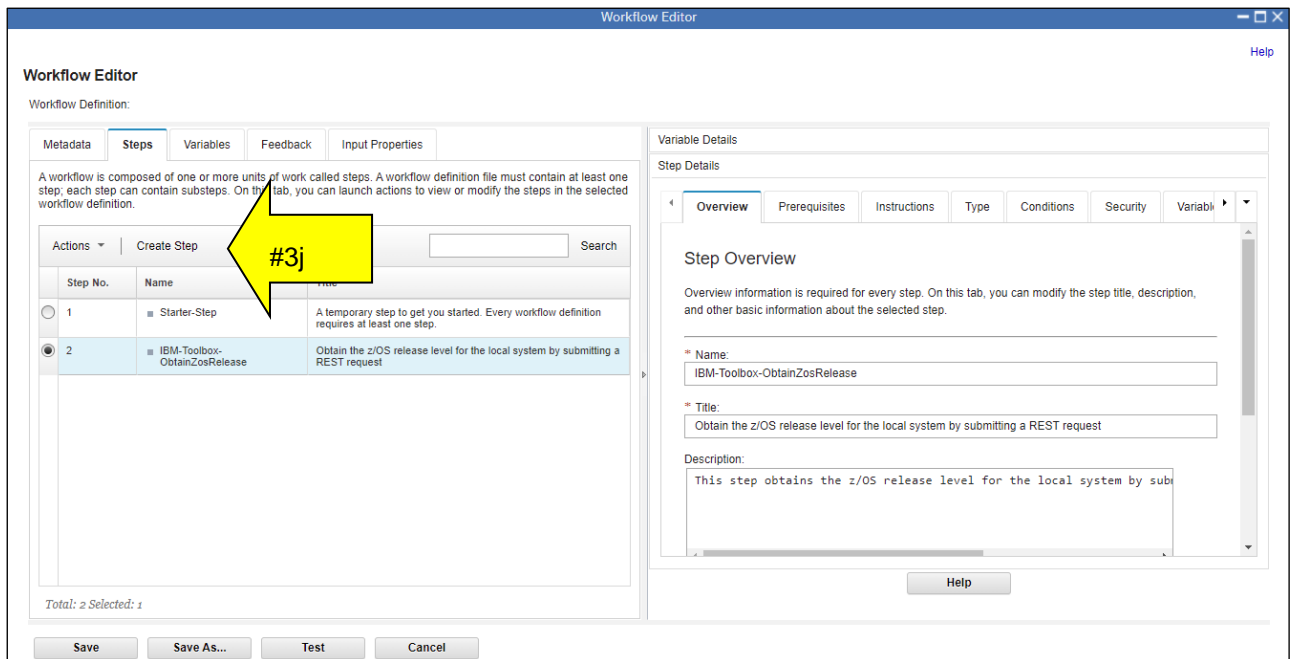
< Back Next > Finish Manage Help

- i. You will then see this. Click **Close**.



Now, let's add another step.

- j. Click on **Create Step** to add another step to the new Workflow.



- k. This step dialog box now appears. You have to decide if you want a **Leaf Step** or a **Parent Step**. **Leaf Step** is selected by default. Select **Parent Step** radio button, so we can try out a Parent Step creation. (The Leaf Step button is the default, so it will be pre-clicked.)
- A **Leaf Step** is simply a step that a user will actual perform.
 - A **Parent Step** is simply a grouping of steps (**Leaf Steps**) that you want to put together. You don't perform a **Parent Step**; you perform a **Leaf Step**.

Create a New Step

☒ Leaf Step ☐ Parent Step

* Name: Enter a unique step name here

* Weight (1-1000): Enter the step weight here

* Title: Enter a title for the step here

Skills: Enter the skills for the step here

Description: Enter a description for the step here

* Step Type: Instructions Only

☐ Optional Step
☐ Auto-Enable
☐ Allow manual marking of step failure

OK Cancel Help

- l. On the **Parent Step** dialog, give your step a **Name**. Of course, there is flexibility for what you can name it, but so that your Workflow matches the Workflow in this lab, call it ***RunStuff***
- The name for a Workflow step must be unique and cannot contain spaces. The user will not see this step name, but it is needed by z/OSMF to know what this step is called in case you wanted to reference it somewhere else.
- m. Provide a **Title**. Again, for simplicity and for matching these lab instructions call it ***Run some z/OS commands***
- Notice that this title can contain spaces. This is what the user will see as the step name on the Workflow. It can be up to 100 characters.
- n. Provide a **Description**. Type whatever you want here.
- You can have a long description that is up to 500 characters.
- o. Click on **OK**. This will add a parent step to your new Workflow under which you can now add Leaf Steps.

The screenshot shows the 'Create a New Step' dialog box. It has two radio buttons at the top: 'Leaf Step' (unselected) and 'Parent Step' (selected). Below these are three input fields: 'Name' with the text 'RunStuff', 'Title' with the text 'Run some z/OS commands', and a 'Description' text area containing the text 'This step will run some z/OS comma'. At the bottom, there is an 'Optional Step' checkbox (unchecked) and three buttons: 'OK', 'Cancel', and 'Help'. Five yellow arrows with labels point to specific elements: '#3l' points to the 'Name' field, '#3m' points to the 'Title' field, '#3n' points to the 'Description' text area, and '#3o' points to the 'OK' button. There is also an unlabeled arrow pointing to the 'Parent Step' radio button.

- p. You will then see this. Click **Close**.



You can see (almost) in the background that your new parent step was created, however, the Workflow Editor now wants you to create a leaf step under the parent step. It presents the **Leaf Step** dialog to you.

- q. Keep the **Leaf Step** radio button selected this time.
- r. Give the leaf step a **Name**. Of course, there is flexibility for what you can name it, but so that your Workflow matches the Workflow in this lab, call it **RunLISTCAT**.
 - Same as for the parent step name, the leaf step name must be unique and cannot contain spaces. The user will not see this step name, but it is needed by z/OSMF to know what this step is called in case you wanted to reference it somewhere else.
- s. Provide a **Title**. Again, for simplicity and for matching these lab instructions call it **Run LISTCAT against a z/OS data set**.
 - Notice that this title can contain spaces. This is what the user will see as the brief step name on the Workflow. It can be up to 100 characters.
- t. Provide a **Description**. Type whatever you want here.
 - You can have a long description that is up to 500 characters.
- u. Provide a **Weight**. Type whatever you want here. Allocating a z/OS data set is easy, so its weight probably isn't a lot.
 - This is how difficult you think that this step is from 1 – 1000. Usually smaller the weight, the easier the step is to do.
- v. Provide a **Skills** description. Type whatever you want here.
 - You can put whatever you think is the right kind of skills it takes to do this step.
- w. Indicate a **Step Type** of **File Template**.
 - A **File Template** will be the JCL that we provide (with some variables in it) from a file already created for this purpose. Traditional JCL would fit into the **File Template** category.
 - There are other **Step Types** that you can explore later. For your very first Workflow, knowing how to submit JCL as a **File Template** can provide a lot of initial value, and is very easy.
- x. When you have a **File Template**, you need to say where it is.
 - Type in **/u/mwalle/zosmf/labs/listcat** if are doing this lab as part of a conference. If you are doing this lab independently you will need to add a file to your z/OS UNIX file system, which you can type in before completing this leaf step (so this step can find the file). The complete file for this step is shown in the Appendix of this lab.
- y. Select **JCL** for **Submit Template As**:
- z. Click on **OK**. This will add a leaf step under the parent step you created.

Create a New Step

☒ Leaf Step ☐ Parent Step

* Name: #3r

* Title: #3s

Description: #3t

☐ Optional Step
☐ Auto-Enable
☐ Allow manual marking of step failure

#3z

* Weight (1-1000): #3u

Skills: #3v

* Step Type: #3w

* Location of file template: #3x
☐ Location contains substitution

* Submit Template As: #3y

You will then see this. Click **Close**.



A wonderful thing to know! The Workflow Editor will never allow you to produce an invalid Workflow. In other words, if you use the Editor you can be assured that a user can create a Workflow that you produced. It might not be what you had intended to create, but that is a different story 😊.

4. Adding more step information to your Workflow.

You've got a parent step with one leaf step under it, and you can see that in Workflow Editor! Let's look at how to add a little more information to that one leaf step at this point.

- a. Under the **Step Details** and **Overview** tab on the right-hand side, browse through what is there. This is the information that you provided on the dialog box when you created the step. But there is a lot more you can do under **Step Details** as you will now find out. Just scan **Overview** for now.

The screenshot shows the Workflow Editor interface. On the left, a table lists steps in a workflow. Step 3.1, named 'RunLISTCAT', is selected. On the right, the 'Step Overview' tab is active, showing details for the selected step. A yellow arrow labeled '#4a' points to the 'Overview' tab, and a pink arrow labeled 'Look' points to the 'Step Overview' section. A yellow arrow labeled '#4b' points to the 'Save' button at the bottom left.

Step No.	Name	Title
1	Starter-Step	A temporary step to get you started. Every workflow definition requires at least one step.
2	IBM-Toolbox-ObtainZosRelease	Obtain the z/OS release level for the local system by submitting REST request
3	RunStuff	Runs some z/OS commands
3.1	RunLISTCAT	Run LISTCAT against a z/OS data set

Step Overview

Overview information is required for every step. On this tab, you can modify the step title, description, and other basic information about the selected step.

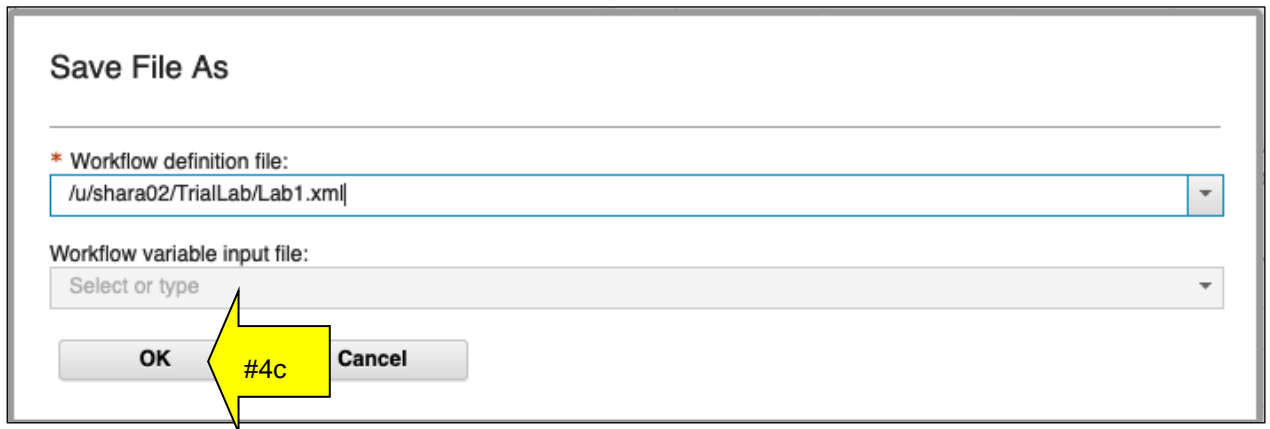
Name: RunLISTCAT

Description: This step will run LISTCAT against a ds.

Buttons: Save, Save As..., Test, Cancel, Help

- b. First though, let's save our work! Click on **Save**. You should save throughout this lab, as you desire. If are able to click on the **Save** button that means you have something valid to save.

- c. Type a location to save your Workflow definition file so far. You must have permission to write the directory, **and** previously that directory must be empty. (The Editor has added a subsequent enhancement later that didn't require an empty directory.) The Editor will create any directories, if they don't already exist. You can see the location where you have saved on the Editor primary panel (circled above). Then click on **OK**.
- If you are doing this lab as part of a conference, use `/u/YourUserId/TrialLab/Lab1.xml` as the location, where *YourUserId* is your assigned userid. Use all lower case for *YourUserId*, such as *share01*. Remember, pathnames are case sensitive.



The image shows a 'Save File As' dialog box. It has a title bar 'Save File As'. Below the title bar, there is a section labeled '* Workflow definition file:' with a text input field containing '/u/shara02/TrialLab/Lab1.xml'. Below this, there is a section labeled 'Workflow variable input file:' with a dropdown menu showing 'Select or type'. At the bottom, there are two buttons: 'OK' and 'Cancel'. A yellow arrow points from the text '#4c' to the 'OK' button.

Let's create some more **Step Details** for our **RunLISTCAT** step. For this portion of the lab, we will be on the right-hand side of the Editor.

- d. Click on **Instructions**. Fill in some helpful documentation that you want the user to see when they are performing this step. Note that in the instructions provided below for our sample Workflow, we are telling the user what input they should be prepared to answer.
 - *What are good instructions?* Here, a user will want to know how to perform this step. For instance, in our Workflow we are running LISTCAT, the user will have to be prepared to supply one variable (a data set name). This is the type of helpful information that good instructions should contain.

The screenshot shows the 'Workflow Editor' window. On the left, the 'Steps' tab is active, displaying a list of steps. Step 3.1, 'RunLISTCAT', is selected. On the right, the 'Step Details' panel is open, and the 'Instructions' sub-tab is selected. A yellow arrow labeled '#4d' points to the 'Instructions' sub-tab. The 'Step Instructions' text area contains the following text:

The user requires detailed instructions for performing the step to completion. On this tab, you can modify the user instructions for the selected step.

These are some good instructions to tell the user how to run this LISTCAT step.

At the bottom of the 'Step Details' panel, there is a 'Help' button. At the bottom of the 'Workflow Editor' window, there are buttons for 'Save', 'Save As...', 'Test', and 'Cancel'.

Next up under **Step Details** for our **RunLISTCAT** step is **Type**. This is an important tab for our sample Workflow because we need to specify some things that we were not able to during step creation.

e. Under **Submit Template As:** select the **JCL** drop down.

- This is saying that the template that we are using (which we said when the step was created, **Path Name of File Template**), is going to be **JCL** for the Workflow to submit. There are other kinds of “work” you do on this step, such as REXX or shell command.

The screenshot shows the 'Workflow Editor' window with the 'Steps' tab selected. The 'RunLISTCAT' step is highlighted in the list. The 'Type' tab is selected in the 'Step Details' section. A yellow arrow points to the 'Submit Template As:' dropdown, which is set to 'JCL'. The 'Location of file template:' field is set to '/u/mwalle/zosmf/labs/listcat'.

Workflow Editor

Workflow Definition: /u/shara02/TrialLab/Lab1.xml

Metadata Steps Variables Feedback Input Properties

A workflow is composed of one or more units of work called steps. A workflow definition file must contain at least one step; each step can contain substeps. On this tab, you can launch actions to view or modify the steps in the selected workflow definition.

Actions Create Step Search

Step No.	Name	Title
1	Starter-Step	A temporary step to get you started. Every workflow definition requires at least one step.
2	IBM-Toolbox-ObtainZosRelease	Obtain the z/OS release level for the local system by submitting a REST request
3	RunStuff	Run some z/OS commands
3.1	RunLISTCAT	Run LISTCAT against a z/OS data set.

Total: 4 Selected: 1

Variable Details

Step Details

Overview Prerequisites Instructions **Type** Conditions Security Variables Feedback

Template Contents

The step type indicates the type of processing that the step performs. On this tab, you can modify the related details.

Step Type: File Template

* Submit Template As: JCL

* Location of file template: /u/mwalle/zosmf/labs/listcat

Location contains substitution

Help

Save Save As... Test Cancel

- f. Scroll down. Under **Max RC**. We want this to be 0, since this step must succeed for the rest of our Workflow steps to work. Put a **0** in that **MAX RC** box. Make **Max LRECL 80**. **note: (LRECL == Record Length)*

The screenshot shows the 'Workflow Editor' window. On the left, the 'Steps' tab is active, displaying a list of steps. Step 3.1, 'RunLISTCAT', is selected. On the right, the 'Step Details' panel is open, showing the configuration for the selected step. A yellow arrow points to the 'Max LRECL' field, which is set to 80. The 'Max RC' field is set to 0. The 'Template Contents' field shows the command '//LISTCAT EXEC PGM=IDCAMS'.

Workflow Editor
Workflow Definition: /u/shara02/TrialLab/Lab1.xml

Metadata | **Steps** | Variables | Feedback | Input Properties

A workflow is composed of one or more units of work called steps. A workflow definition file must contain at least one step; each step can contain substeps. On this tab, you can launch actions to view or modify the steps in the selected workflow definition.

Step No.	Name	Title
1	Starter-Step	A temporary step to get you started. Every workflow definition requires at least one step.
2	IBM-Toolbox-ObtainZosRelease	Obtain the z/OS release level for the local system by submitting a REST request
3	RunStuff	Run some z/OS commands
3.1	RunLISTCAT	Run LISTCAT against a z/OS data set.

Total: 4 Selected: 1

Step Details
Overview | Prerequisites | Instructions | **Type** | Conditions | Security | Variables | Feedback

* Submit Template As: JCL

* Location of file template: /u/mwalle/zosmf/labs/listcat

☐ Location contains substitution

Max RC: 0

Max LRECL: 80

Template Contents: //LISTCAT EXEC PGM=IDCAMS

Help

Save Save As... Test Cancel

- g. Keep scrolling in **Step Details** to the **Template Contents**. This was pulled in from the **Path Name of File Template** location during step creation and you can see it here. Notice that there are variables that need to be replaced with “real values”, For that reason, you must check **Contains variable substitution**. This is important, or the JCL would try to run as you see it and it would fail!

Note that if you don't see anything in the **Template Contents**: that means you didn't type the right template name in the **Path Name of File Template** location. You should see the information below.

Variable Details

Step Details

Overview Prerequisites Instructions **Type** Conditions Security Variable

80

Template Contents:

```
//LISTCAT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
LISTCAT ALL ENT($instance-DSN)
```

☒ Contains variable substitution

[Open in z/OSMF Desktop Editor](#)

Help

Nothing else on this tab needs to be changed. We don't need to do anything in the **Conditions** or **Security** tabs.

5. Variables in your Workflow.

We have variables in our Workflow steps, if you looked closely in the JCL template. That is how you bring value to Workflow steps and allow the user to give you some input. You put that input into a variable and then substitute it in the template when it is time to run the JCL and is then submitted by the Workflow.

You can provide variables to a Workflow in two ways: you can individually create them here under **Variables** tab from the Workflow Editor, or you can pull them in from an external file (called “Input Properties”) that has them already resolved. We will not explore Input Properties in this lab.

Let’s create variables in the Editor:

- Click on the **Variables** tab.
- Then, click on **Actions** drop down, and select **Create Variable**. Alternatively, you could also click on **Create Variable** right next to **Actions**.

Workflow Editor

Workflow Definition: /u/shara10/TrialLab/Lab11.xpf

Variables can be used for substitution in workflow step instructions and templates. On this tab, you can launch actions to view and modify the variables in the selected workflow definition.

Actions **Create Variable** Search

Create Variable **#5b**

Name	Scope	Category
zos_version	instance	IBM PROVIDED

Total: 1 Selected: 1

Save Save As... Test Cancel

- c. The **Create a New Variable** dialog opens up. Here's where we will define two variables that we will use in our sample Workflow. Fill in the values you see here, so that the lab is consistent with the screen shots. Enter **DSN** for **Variable Name**. You cannot use blanks in your variable names.
- d. For Scope, select **Instance**. This means that the scope of the variable will be inside this instance of the Workflow. The other option is **Global**, which means that the variable can be used outside this instance.
- e. For **Label** type **Data set to run LISTCAT against**. This will be the short name that the Workflow user will see as a description of what to fill in here.
- f. For **Abstract** type what the user will see from the Workflow, to know what this field will be used for.
- g. For **Description** type a longer description of what the user might need to know about this variable.
- h. For **Category** select **General**.
- i. For **Visibility** select **public**. This means that the user will see the variable while doing the Workflow. The other option is **private**, meaning that users won't see this variable in the Workflow.
- j. For **Type** select **string**. The data set name is a string variable. There are other options for other kinds of variables. This will help the Workflow validate what the user must enter. Later, we'll make it more specific kind of string.
- k. Click on **Next>**.

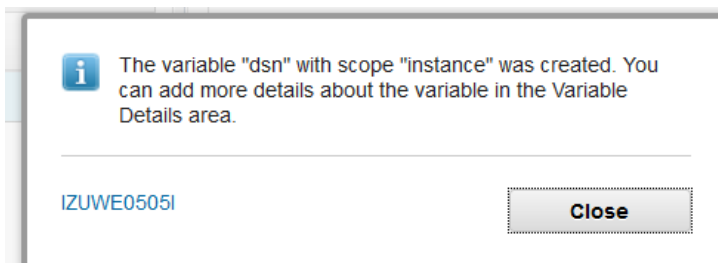
The screenshot shows the 'Create a New Variable' dialog box. It has two columns of fields. The left column contains: '* Variable Name' (text box with 'DSN'), 'Scope:' (dropdown with 'instance'), '* Label:' (text box with 'Data set to run LISTCAT against'), '* Abstract:' (text box with 'LISTCAT DSN'), and '* Description:' (text area with 'e of the data set to run LISTCAT aga'). The right column contains: '* Category:' (dropdown with 'General'), 'Visibility:' (dropdown with 'public'), '* Type:' (dropdown with 'string'), and two checkboxes: 'Prompt At Create' and 'Required At Create'. At the bottom are four buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. Yellow arrows with labels point to the following elements: #5c points to the Variable Name field; #5d points to the Scope dropdown; #5e points to the Label field; #5f points to the Abstract field; #5g points to the Description text area; #5h points to the Category dropdown; #5i points to the Visibility dropdown; #5j points to the Type dropdown; and #5k points to the 'Next >' button.

- l. Continuing on in the dialog. Add a **Default Value** that you want for this data set, if the user doesn't supply one. Type **MY.DS.NAME** here, because that is the default name we want to use for this sample.

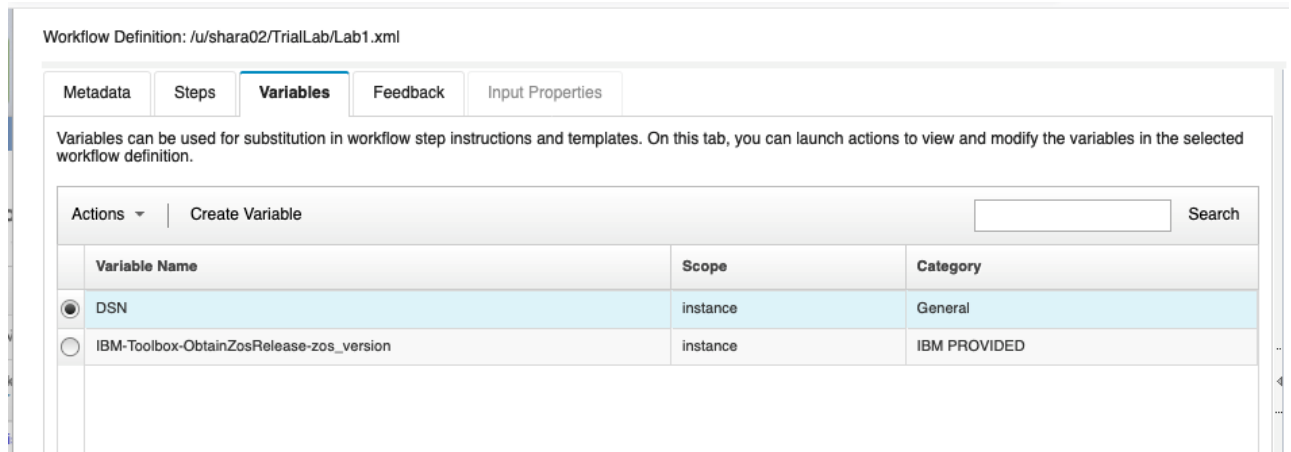
- m. Under **Validation Criteria**, select **Validation Type**. This tells the Workflow that you want to validate the input from the user for this variable.
- n. Then, you say that that the validation should be for a data set name. Under **Validation Option**, select **DSNAME**. This will make the Workflow make sure the user types in a valid z/OS data set name when overriding the default.
- o. Finally, click on **Finish**.

The screenshot shows the 'Create a New Variable' dialog box. It has a title bar and a main content area. The content area includes a 'Default Value' field with the text 'MY.DS.NAME' and a yellow arrow labeled '#5l' pointing to it. Below this is a checkbox labeled 'Check for multi-line'. The 'New Value Choice' field is empty, with a yellow arrow labeled '#5i' pointing to it. To the right of this field is an 'Add Choice' button. Below the 'New Value Choice' field is a list box for 'Existing Value Choices', which is empty. To the right of this list box are three buttons: 'Set Default', 'Remove Choice', and 'Add Choice'. Below the list box is a checkbox labeled 'Value Must be a Choice'. The 'Validation Criteria' section has two radio buttons: 'Validation Type' (selected) and 'Expression'. A yellow arrow labeled '#5m' points to the 'Validation Type' radio button. The 'Validation Options' section has a dropdown menu with 'DSNAME' selected. A yellow arrow labeled '#5n' points to the dropdown menu. At the bottom of the dialog are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. A yellow arrow labeled '#5o' points to the 'Finish' button.

You now created a Workflow variable. Click **Close**



When you are done, you should see two variables (***DSN*** and ***IBM-Toolbox-ObtainZosRelease-zos_version***) under the **Variables** tab.



An important thing to know about variables and Workflows:

Let's get back to that important check mark we used in our JCL template **Contains variable substitution**. Workflows do variable substitution, as you could have guessed, but how does it do it?

It uses a mechanism provided by the open-source Velocity engine created by the Apache Velocity Project. It can be pretty powerful and allow conditional directives, but we'll stick to the simplest form for our lab, which is straight substitution. How do you invoke the Velocity engine? By simply coding ***\$instance-variable*** in your template where you want it. For instance, in our JCL template it is coded:

\$instance-DSN

which the Workflow will substitute in the JCL to simply:

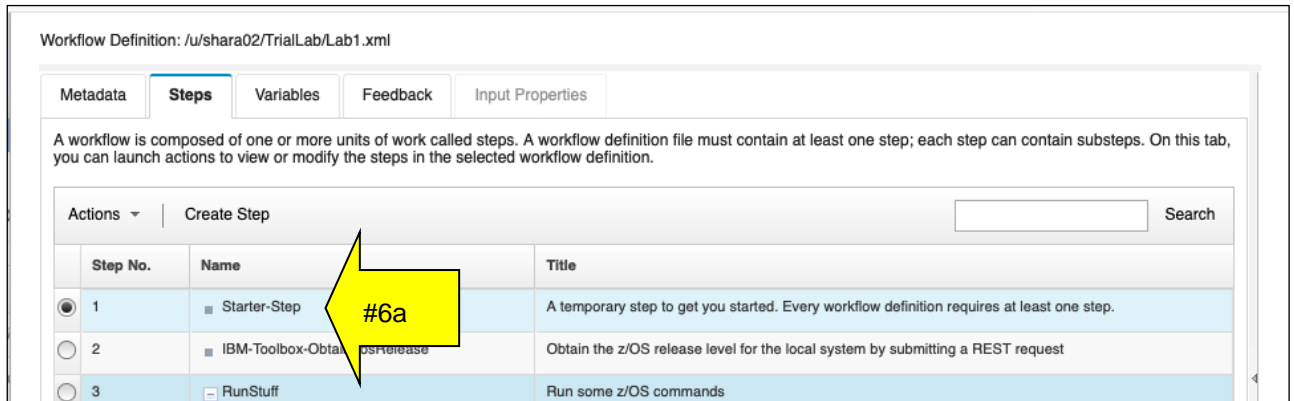
MY.DS.NAME (the default) or **MWALLE.LAB1.DSN** as appropriate.

6. Putting Variables in your Steps.

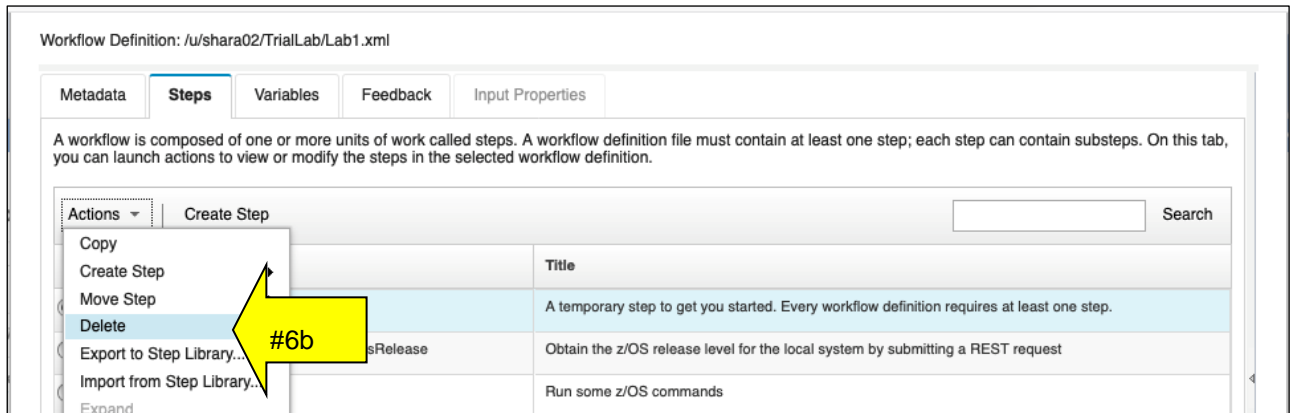
At this point, we have added one step (really one parent and one leaf step) and two variables in our Workflow. However, we don't have the variables *associated with any steps*. We need to put the variables in a step.

First, though, let's clean up our Workflow. Every new Workflow will have a starter step (called **Starter-Step**), which we can see at the first step in the Workflow. Let's delete that starter step, because we don't need it. We have to have that step, as a Workflow *must* have a step if it is valid, and the Editor will only allow you to have a valid Workflow. Therefore, the Editor gives you a valid step, but it is expected that you'll delete it.

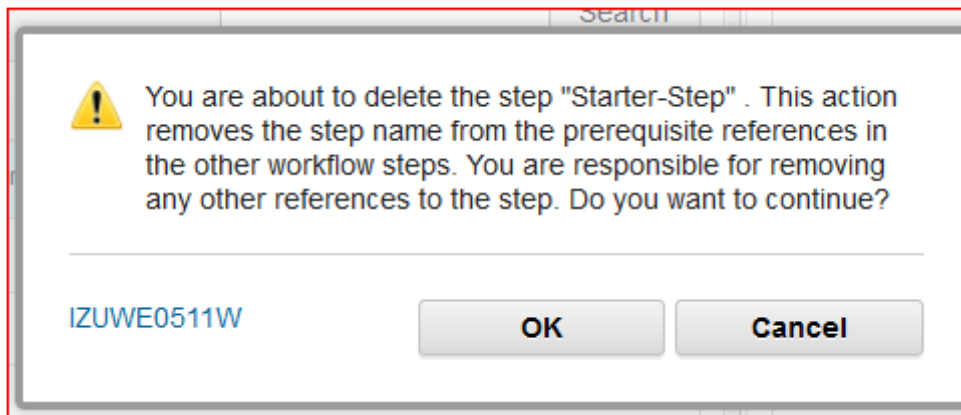
- a. Click on the **Starter-Step**, which is step #1.



- b. From **Actions**, select **Delete** from the drop down.

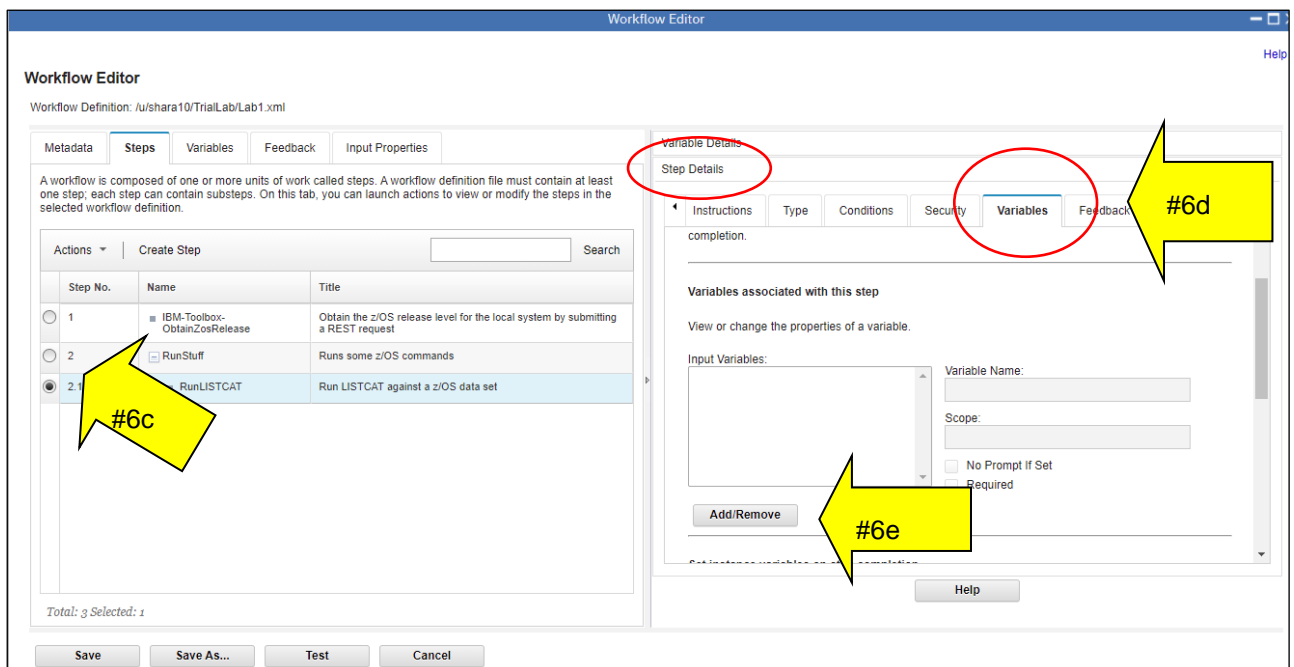


You will see the confirmation, click on **OK**. You now know how to delete a step in a Workflow you don't want.



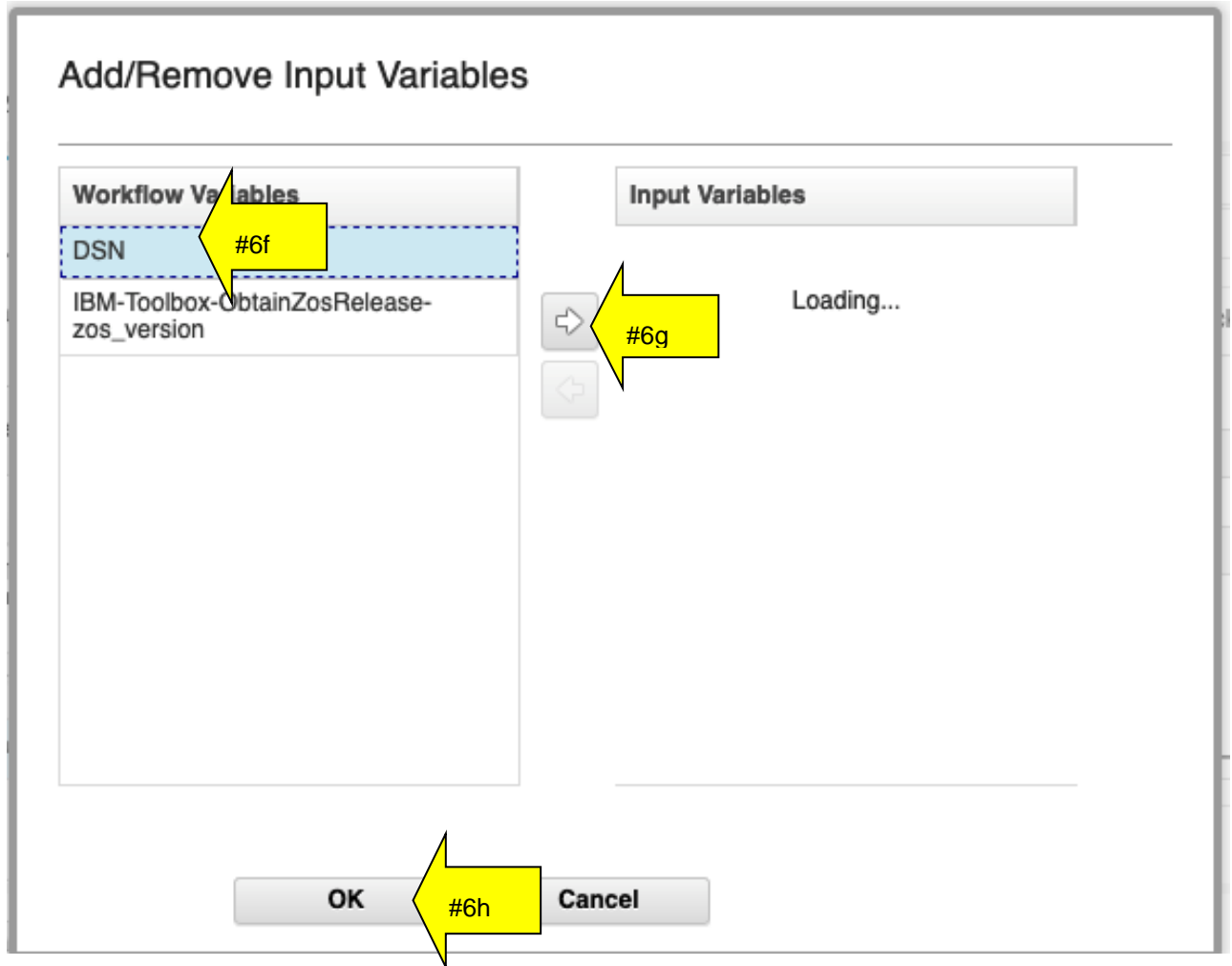
In your Workflow, you have one step you imported, and one parent step with one leaf step. Let's associate that leaf step with the variable that we have defined.

- c. Click on the leaf step **RunLISTCAT**, which is now step #2.1. (You might have to untwist the parent step to see it.)
- d. Click on the right-hand side's **Step Details**, and then **Variables**.
- e. Scroll down to the bottom, so you can see the **Add/Remove** button. Click on **Add/Remove**.

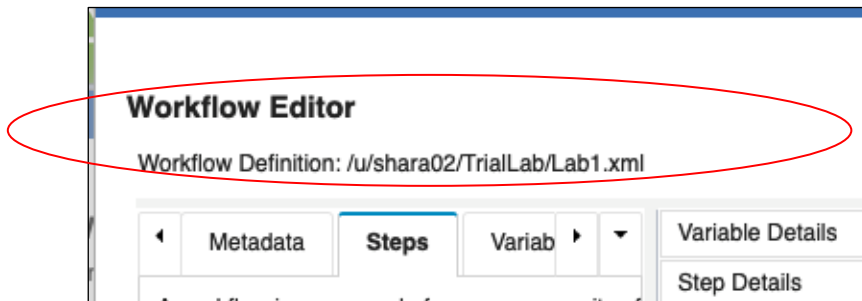


The **Add/Remove Input Variables** dialog appears. Look, there are the two variables, one added by the import step and one that we added. We need to move the one we added from the **Workflow Variables** column into the **Input Variables** column.

- f. Click on **DSN**.
- g. Then the **arrow** to move it to the other column.
- h. Click on **OK** to complete the dialog.



- i. Let's save the Workflow and see what we've created at this point. Back on the main Editor screen do a **Save**. The save will put your Workflow definition file into your existing directory as shown by **File Path:** (which was `/u/YourUserId/TrialLab/Lab1.xml` remembering to get the case correct.)

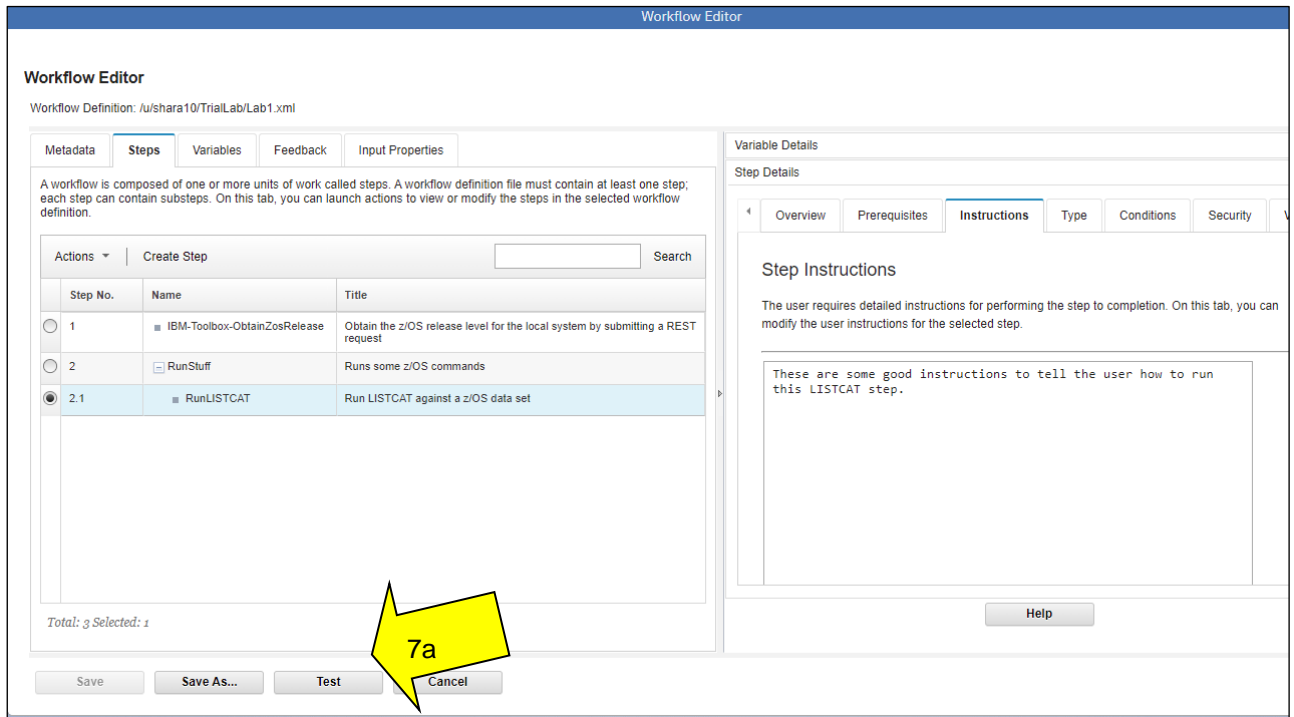


7. Trying out your Workflow.

You have a very basic Workflow completed. Let's try it out to see what it looks like to a user!

It is easy to try out your Workflow that you are working on with the **Test** button at the bottom.

- a. Click on **Test** to create a Workflow instance to see how your Workflow looks, which will launch you into Workflows.



- b. On the **Create Workflow** dialog, verify the name of your Workflow definition file you just created. (which was `/u/YourUserId/TrialLab/Lab1.xml`) where *YourUserId* was the userid you were given for the lab.
- c. Ensure the Location is **SHARPLEX.S2** for this lab system.
- d. Click on **Next**.

The screenshot shows the 'Create Workflow' dialog box. It contains three input fields and a set of navigation buttons at the bottom. Yellow arrows with labels point to specific elements:

- Arrow **7c** points to the 'Location (system) of definition and variable input files:' dropdown menu, which currently shows 'SHARPLEX.S2 (S2)'.
- Arrow **7b** points to the 'Workflow definition file:' dropdown menu, which currently shows '/u/shara10/TrialLab/Lab1.xml'.
- Arrow **7d** points to the 'Next >' button.

The 'Workflow variable input file:' field is currently empty and shows 'Select or type'. The navigation buttons at the bottom are '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

- e. Continue by filling in the **Workflow name**: Notice that this field is prefilled in with what you specified in the Editor, **Metadata** screen, as the **Description**. It may be that you see an error for this **Workflow name**. Currently, the Editor accepts characters for the **Description** that aren't valid for **Create Workflow** (like "/"). This is a known situation and easy to get past. Just remove the / and type a unique name for your new Workflow during creation (probably just removing the "/"). For instance, use: **YourUserid LAB1: This is my first Workflow created by the zOSMF Workflow Editor**. Notice that **-Workflow_0** is added by the Workflow creation process, and you did not have in your Editor **Metadata Description**. Be aware, you'll need to change the default name and Workflow ID each time you create a new Workflow definition (or you'll see an error).
- f. Click on **Assign all steps to owner user ID**, so that you can move the verification of this Workflow quickly. This is a shortcut we'll use to rapidly move into testing our Workflow.
- g. Then click on **Finish**.

The screenshot shows the 'Create Workflow' form with the following fields and annotations:

- Location (system) of definition and variable input files:** SHARPLEX.S2
- Workflow definition file:** /u/shara01/TrialLab/Lab1.xml
- Description:** Now using the z/OSMF Workflow Editor.
- Vendor:** SHARA01 Corporation Inc. **Version:** 1.0 **Is Callable:** ? Cannot be called by another workflow
- * Workflow name:** SHARA01 LAB1: This is my first Workflow Created by the zOSMF Workflow Editor (Annotation 7e points to this field)
- * Owner user ID:** shara01 **Archive SAF ID:** ? shara01 *** System (where workflow steps will be performed):** SHARPLEX.S2
- Comments:** (Empty text area)
- * Access(Learn More):** Public
- ☐ Save jobs output
- ☒ Open workflow on finish ☒ Assign all steps to owner user ID (Annotation 7f points to this checkbox)
- ☐ Delete workflow on completion
- Buttons:** < Back, Next >, Finish (Annotation 7g points to this button), Cancel, Help

Now, this should look familiar. You can see a stand-alone leaf step called “Obtain the z/OS release level for the local system by submitting a REST request” and a Parent step call “Run some z/OS commands” with 1 leaf step called “Run LISTCAT against a z/OS data set.” (The Workflow Editor **Step Name** isn’t shown in the Workflow.) The Skill Category is what you put before (such as “Any z/OS user”, as shown below). These are the values that we used in the Editor called **Steps->Title** and **Steps->Skills**.

- h. Click on the blue **Obtain the z/OS release level for the local system by submitting a REST request**, so we can try out one of leaf steps we created.

The screenshot shows the 'Workflows' window with the following details:

- Workflow Details:**
 - Description: Now using the z/OSMF Workflow Editor.
 - Owner: shara02
 - System: SHARPLEX.S2
 - Is Callable: Cannot be called by another workflow
 - Contains Parallel Steps: No
 - Percent complete: 0%
 - Status: In Progress
 - Access (Learn More): Public
 - Workflow definition file: /u/shara02/TrialLab/Lab1.xml
 - System of workflow definition file: SHARPLEX.S2
 - Delete workflow on completion: No
- Workflow Steps:**

State	No.	Title	Call Workflow	Automated	Owner	Skill Category	Assignees
Ready	1	Obtain the z/OS release level for the local system by submitting a REST request		Yes	shara02		shara02
In Progress	2	Run some z/OS commands					
Ready	2.1	Run LISTCAT against a z/OS data set.		No	shara02	Any z/OS User	shara02

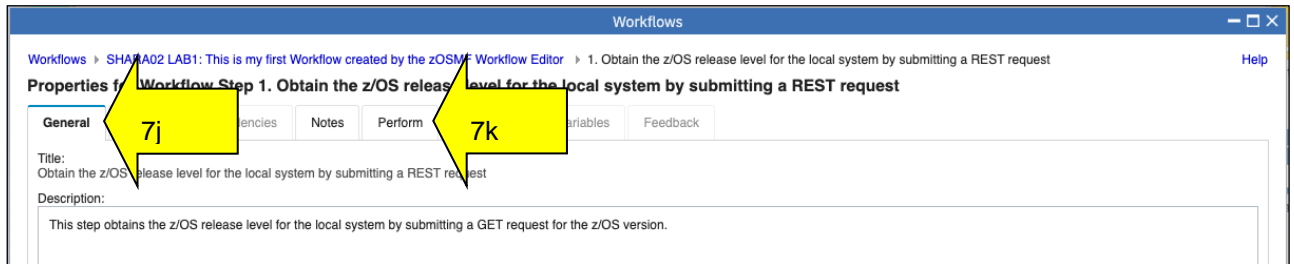
Annotations in the image include a red circle around the title of step 1, a yellow arrow labeled '7h' pointing to it, and a red circle around the 'Any z/OS User' skill category for step 2.1.

Notice that at any time you can refer back to your Workflow definition, by just clicking on the z/OSMF window for **Workflow Editor**, and then return to your Workflow by clicking back to the **Workflows** window. This is a handy way to compare what you put in the Editor, and what it looks like in a real Workflow for a user.

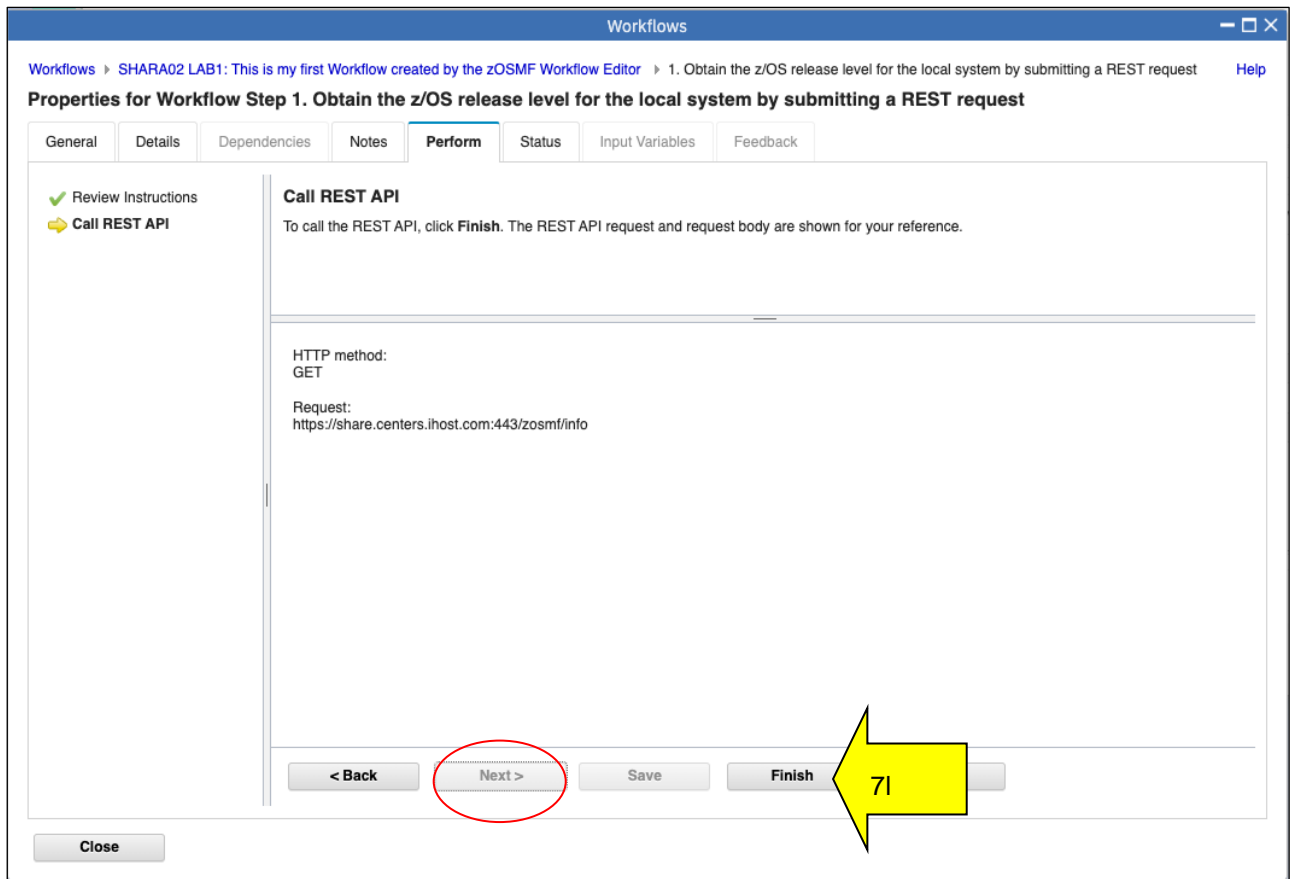
The diagram shows two overlapping windows: 'Workflow Editor' on top and 'Workflows' on the bottom. Two large pink arrows indicate the navigation path: one arrow points from the 'Workflow Editor' window down to the 'Workflows' window, and another arrow points from the 'Workflows' window up to the 'Workflow Editor' window, illustrating how to switch between the two views.

From the Workflow application, you can see the step information you provided from the Editor here.

- j. Look at the information on the **General** tab (Editor input was: **Steps -> Step Details -> Step Overview -> Description**)
- k. Go to the **Perform** tab.



- I. To Perform the step, since no variables are required you can keep hitting the “**Next**” until it is grayed out. Click “**Finish**” to execute the command.



m. Once the job finishes, you can look at the status to ensure that it executed fine. Click **Close**.

The screenshot shows the 'Workflows' application window. The title bar reads 'Workflows'. The breadcrumb navigation is 'Workflows > SHARA02 LAB1: This is my first Workflow created by the zOSMF Workflow Editor > 1. Obtain the z/OS release level for the local system by submitting a REST request'. A 'Help' link is in the top right. The main heading is 'Properties for Workflow Step 1. Obtain the z/OS release level for the local system by submitting a REST request'. Below this is a tabbed interface with 'General', 'Details', 'Dependencies', 'Notes', 'Perform', 'Status', 'Input Variables', and 'Feedback'. The 'Status' tab is selected. It shows 'State: Complete' with a green checkmark, 'Expected status code: 200', and 'Actual status code: 200'. Below this are three sub-tabs: 'Request', 'Response', and 'Message'. The 'Response' tab is selected, displaying a JSON object in a text area. The JSON contains an 'api_version' of '1' and a list of 'plugins' with their default names, statuses, and versions. A 'Close' button is at the bottom left of the window.

Workflows

Workflows > SHARA02 LAB1: This is my first Workflow created by the zOSMF Workflow Editor > 1. Obtain the z/OS release level for the local system by submitting a REST request [Help](#)

Properties for Workflow Step 1. Obtain the z/OS release level for the local system by submitting a REST request

General Details Dependencies Notes Perform **Status** Input Variables Feedback

State: **Complete** Expected status code: 200 Actual status code: 200

Request **Response** Message

```
{
  "api_version": "1",
  "plugins": [
    {
      "pluginDefaultName": "z\\OS Operator Consoles",
      "pluginStatus": "ACTIVE",
      "pluginVersion": "HSMA250;PH38968;2021-11-17T02:45:42"
    },
    {
      "pluginDefaultName": "Software Deployment",
      "pluginStatus": "ACTIVE",
      "pluginVersion": "HSMA254;PH40045P;2021-12-07T18:15:54"
    },
    {
      "pluginDefaultName": "Variables",
      "pluginStatus": "ACTIVE",
      "pluginVersion": "HSMA250;PH41196;2021-12-15T13:52:24"
    },
    {
      "pluginDefaultName": "Workflow",
      "pluginStatus": "ACTIVE",
      "pluginVersion": "HSMA257;PH40841;2021-11-09T03:08:20"
    },
    {
      "pluginDefaultName": "IncidentLog",
      "pluginStatus": "ACTIVE"
    }
  ]
}
```

Close

- n. You should see that the step is Complete. Go ahead and try running the leaf step “Run LISTCAT against a z/OS data set.”. When prompted for a Data set to run LISTCAT against. You can use the data set <YOUR ID>.ZFS.USER. Ex: SHARA02.ZFS.USER

Workflows

SHARA02 LAB1: This is my first Workflow created by the zOSMF Workflow Editor

SHARA02 LAB1: This is my first Workflow created by the zOSMF Workflow Editor

Notes | History

Workflow Details

Description: Now using the z/OSMF Workflow Editor.	Owner: shara02	System: SHARPLEX.S2
Is Callable: Cannot be called by another workflow	Contains Parallel Steps: No	Percent complete: 50%
Steps complete: 1 of 2	Status: In Progress	Access(Learn More): Public
Workflow definition file: /u/shara02/TrialLab/Lab1.xml	System of workflow definition file: SHARPLEX.S2	Delete workflow on completion: No

Workflow Steps

Actions Search All step content

No filter applied

State	No.	Title	CalledWorkflow	Automated	Owner	Skill Category
Complete	1	Obtain the z/OS release level for the local system by submitting a REST request		Yes	shara02	
In Progress	2	Run some z/OS commands				
Ready	2.1	Run LISTCAT against a z/OS data set.		No	shara02	Any z/OS User

Total: 3 Selected: 1

Return to Workflows Refresh Last refresh: Jul 20, 2022, 3:46:14 PM local time (Jul 20, 2022, 7:46:14 PM GMT)

This concludes the most basic part of the lab. You have learned how to:

1. Create a brand-new Workflow
2. Import a step from the Step Library
3. Add variables to the Workflow, so that the user can provide unique information.
4. Add a parent step and a leaf step.
5. Associate the variables with the Workflow steps.
6. Run JCL in batch from a Workflow. (And how to code that JCL for the Velocity engine variables)
7. Save your Workflow definition file.
8. Know how to associate the Workflow Editor fields with what the user sees in a Workflow.

If you have enough time and want to continue, we will now proceed with... putting steps into a library for future import to other Workflows (i.e. reusing steps).

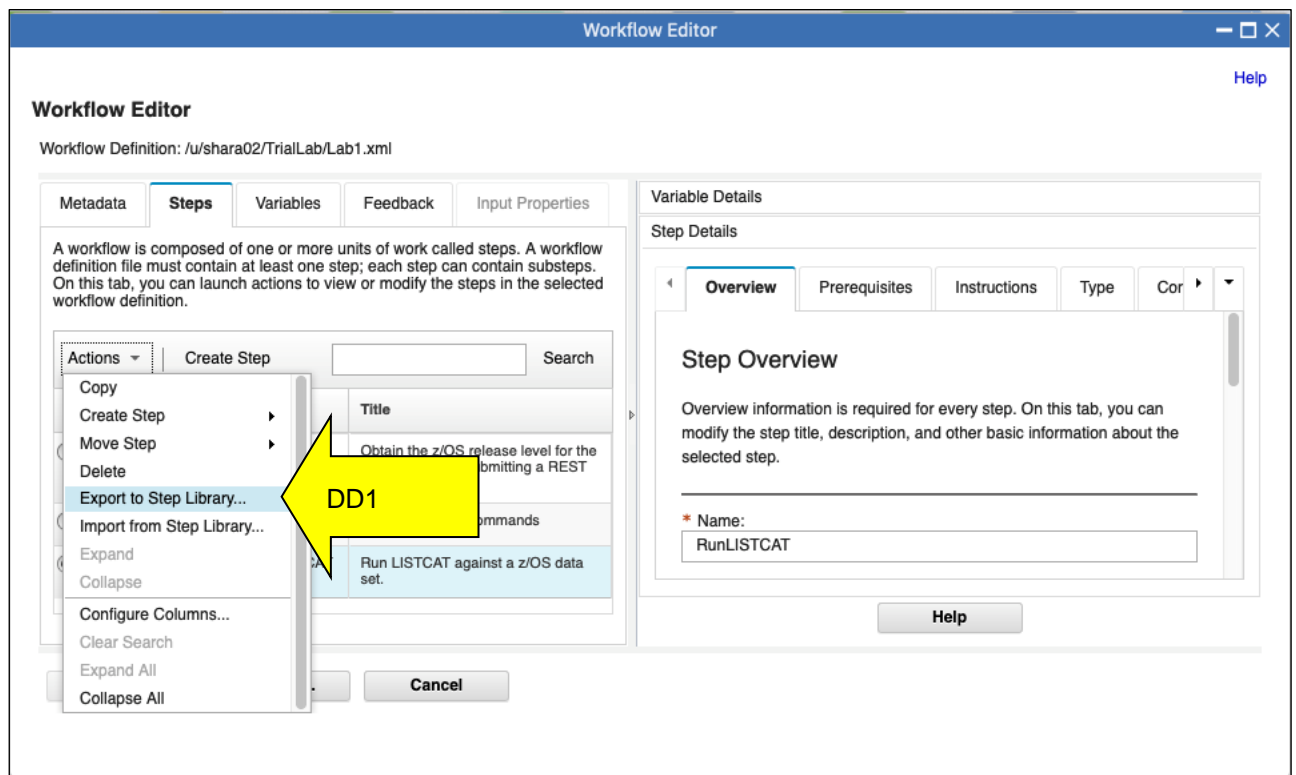
Deeper Dive

Putting steps into a library for future use in other Workflows

One of the nicest things available in the Workflow Editor, is that you can use the Editor to build a step, export it, and then import it into another Workflow you are building. This is a very nice way of reusing common steps that you need in several Workflows. We just saw how you could create a step that allocated a data set, which is a common activity. Let's now work on exporting that step, and then importing it into another Workflow.

DD1: Make sure you still are in the Editor and editing the Workflow definition file `/u/YourUserid/TrialLab/Lab1.xml`.

Click on **Steps**, and click on **step 2.1 (RunLISTCAT)** so we can export that step into our Step Library for other Workflows. Then click on **Actions** pulldown, and then select **Export to Step Library...**



DD2: You can now name the step in the library so others can find it. Because there are multiple people doing this lab, name the step something unique so you can find the specific one you export, and, so you don't have two steps in the same Workflow with the same name for this lab. Use something like **YourUserid_RunLISTCAT**. As we've seen before, the name cannot contain blanks.

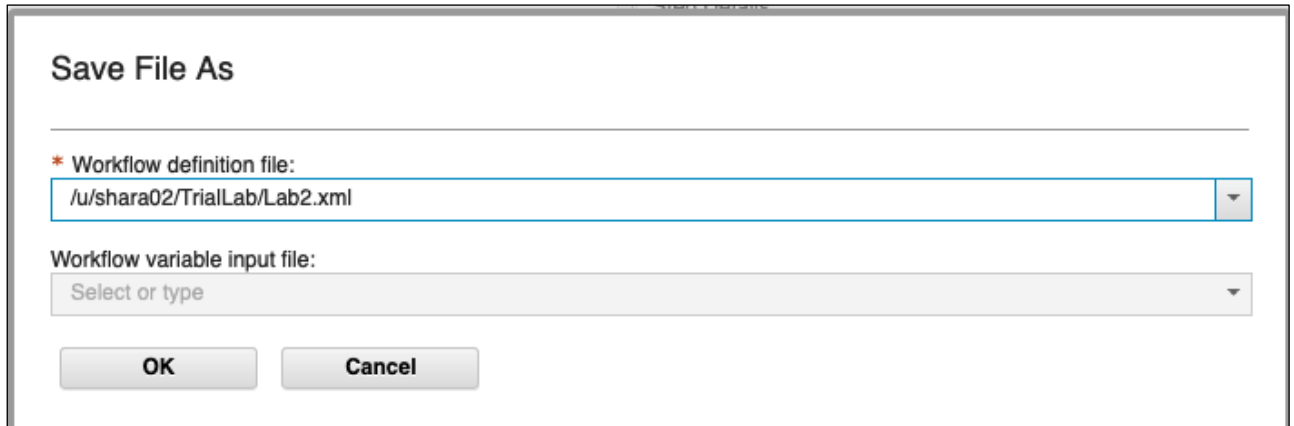
Also, type in a **Description** that you think will be helpful to other Workflow designers about what your exported step does.

Then click on **OK**.

The screenshot shows a dialog box titled "Export to Step Library". It has two main input fields: "Name:" and "Description:". The "Name:" field contains the text "SHARA02_RunLISTCAT". A yellow arrow points to this field with the text "DD2" next to it. The "Description:" field contains the text "This step will run LISTCAT against a z/OS data set.". At the bottom of the dialog box, there are three buttons: "OK", "Cancel", and "Help".

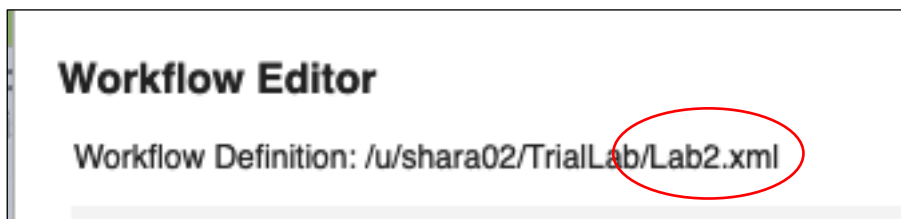
That was it! You have your step ready to share for others.

DD3: For the purposes of this lab, let's save our existing lab (Lab1.xml) into another Workflow definition file (called Lab2.xml) to perform our advance techniques on. Do a **Save As...** and type **/u/YourUserID/TrialLab/Lab2.xml**. Click on **OK**.



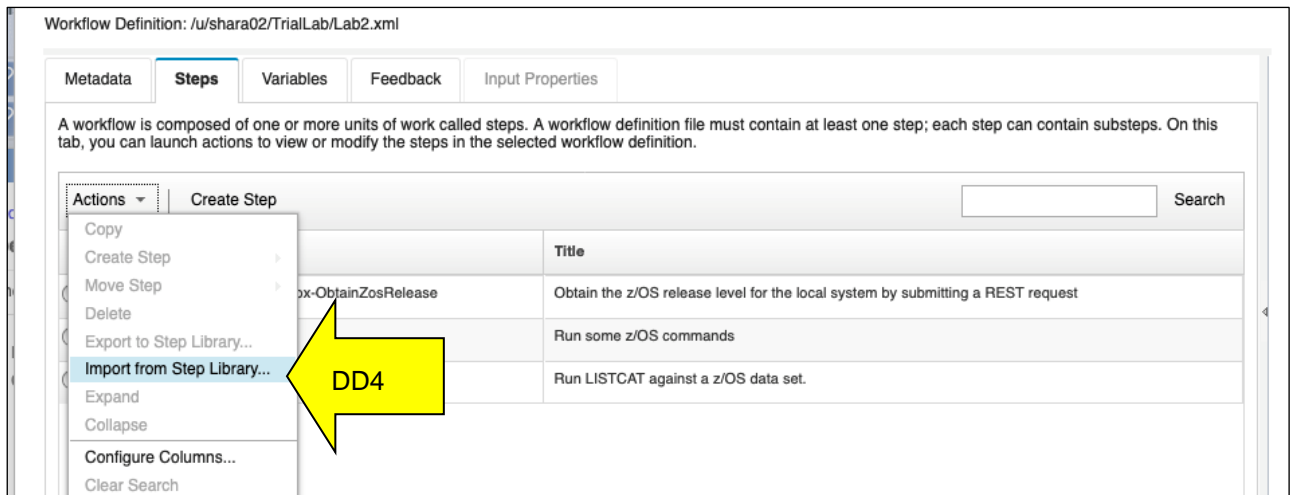
The image shows a 'Save File As' dialog box. At the top, the title is 'Save File As'. Below the title is a horizontal line. Underneath, there is a label '* Workflow definition file:' followed by a text input field containing the path '/u/shara02/TrialLab/Lab2.xml'. Below this is another label 'Workflow variable input file:' followed by a dropdown menu with the text 'Select or type'. At the bottom of the dialog are two buttons: 'OK' and 'Cancel'.

Verify that you saved into the right spot, so you are not losing or overlaying any of your work in Lab1.xml. Check the **File Path:** name at the top to make sure you are *now* in **Lab2.xml**.



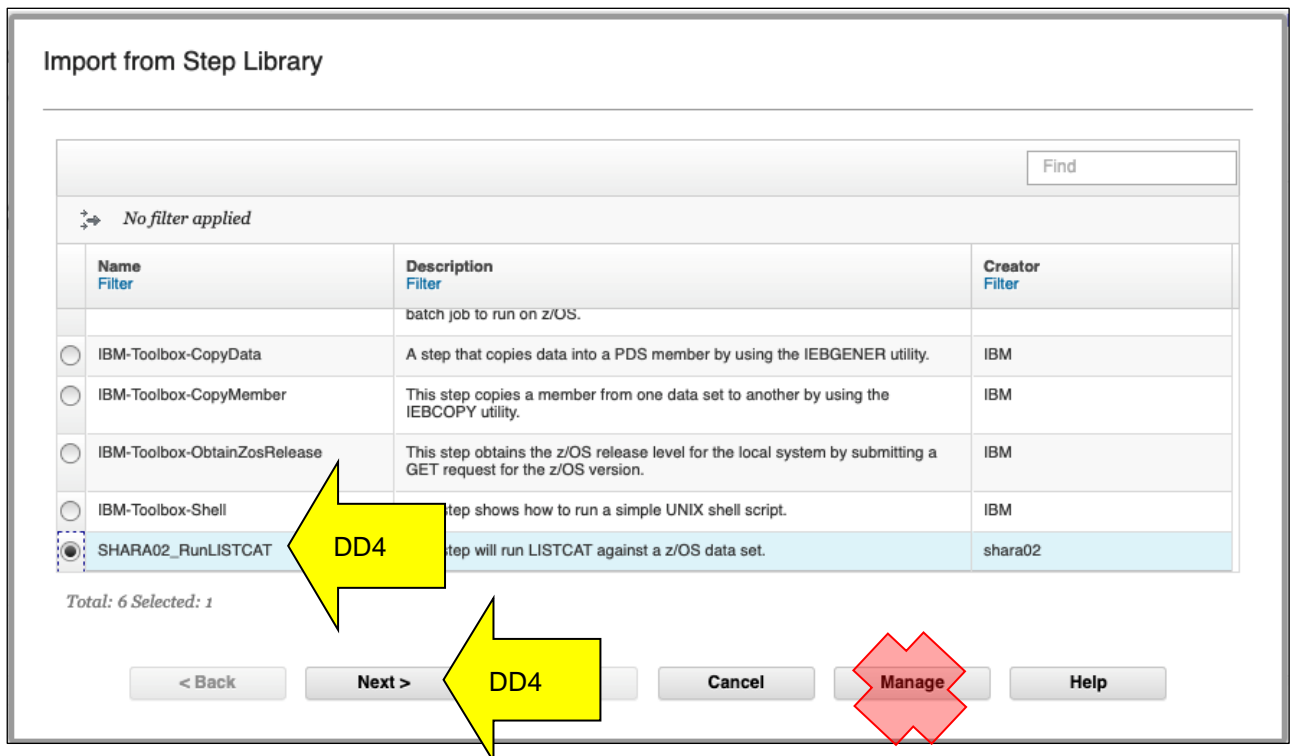
The image shows a 'Workflow Editor' window. The title is 'Workflow Editor'. Below the title, it says 'Workflow Definition: /u/shara02/TrialLab/Lab2.xml'. The path '/u/shara02/TrialLab/Lab2.xml' is circled in red.

DD4: Under the Steps tab, click on Actions, Import from Step Library. (This should be familiar to you now)



You can see the steps that are in the Step Library. There might be more than what is shown below, but you should notice your step that you exported above. Click on your exported step name (**YourUserID-RunLISTCAT**), and then **Next**. If you don't see **Next**, you might have to resize the window.

As an aside point: if you wanted to delete a step from the Step Library, you *would* click on **Manage**, and then under Actions select **Delete** to delete your own steps.



Then click on **Next>**. Which brings you to the follow. Click on **Finish**.

Import from Step Library

* Name:
SHARA02_RunLISTCAT

< Back

Next >

Finish

DD4

Manage

Help

Success! Click on **Close**.

 The leaf step "SHARA02_RunLISTCAT" was imported into the workflow definition.

IZUWE0402I

Close

DD5: Now that we have added the step into our workflow, lets clean up the step we created from the Step Library. Go back into the Step Library (Actions -> Import from Step Library) and click **Manage**.

Import from Step Library

Find

No filter applied

	Name Filter	Description Filter	Creator Filter
<input type="radio"/>	IBM-Toolbox-CreateDataset	A step that allocates (creates) a partitioned data set (PDS) by submitting a batch job to run on z/OS.	IBM
<input type="radio"/>	IBM-Toolbox-CopyData	A step that copies data into a PDS member by using the IEBGENER utility.	IBM
<input type="radio"/>	IBM-Toolbox-CopyMember	This step copies a member from one data set to another by using the IEBCOPY utility.	IBM
<input type="radio"/>	IBM-Toolbox-ObtainZosRelease	This step obtains the z/OS release level for the local system by submitting a GET request for the z/OS version.	IBM
<input type="radio"/>	IBM-Toolbox-Shell	This step shows how to run a simple UNIX shell script.	IBM

Total: 6 Selected: 0

< Back

Next >

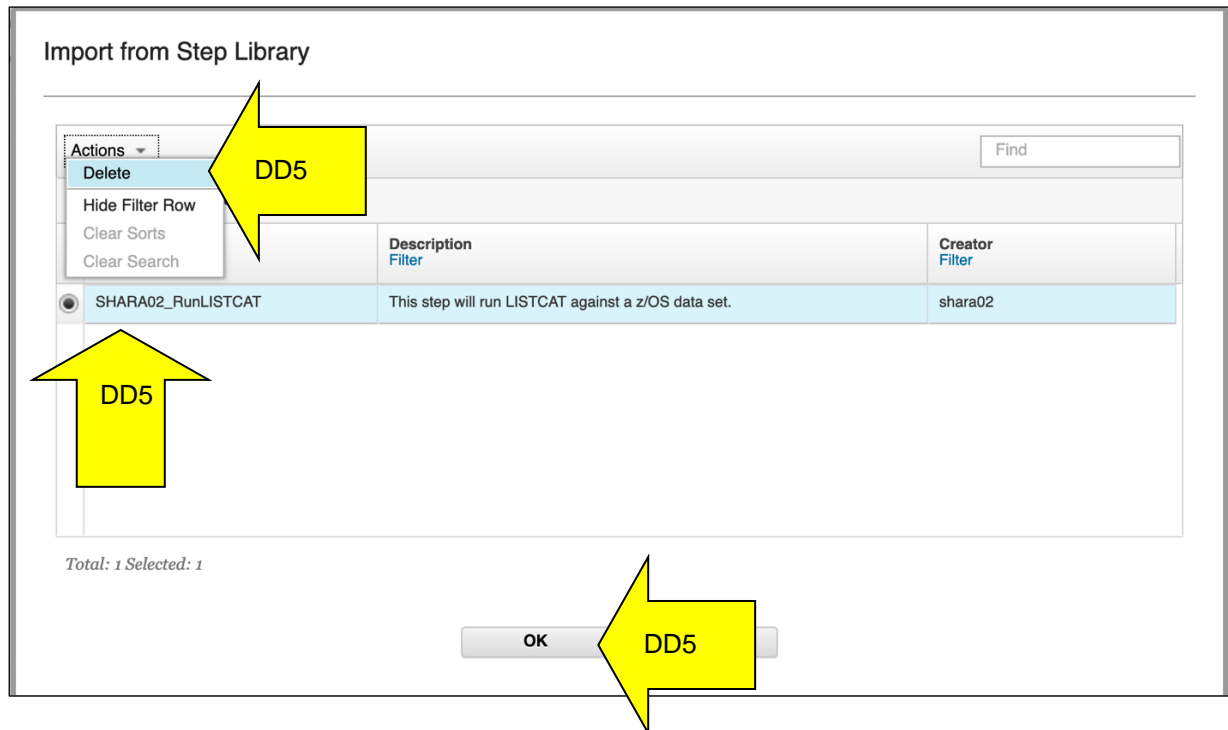
Finish

Cancel

Manage

DD5

On the next window you should see the step you created. Go ahead and select that step, then click on **Actions -> Delete**. You will see the step removed.. Go ahead and click **OK**, and **Cancel** in the Step Library Window.



Take a step back...what happens when you first open a workflow definition?

Overview of Marshalling and Unmarshalling of Workflow files: When you open a workflow definition file for editing, the Workflow Editor locates the file and reads it into storage. The Workflow Editor also identifies any associated files that are referenced in the workflow definition and reads them into storage.

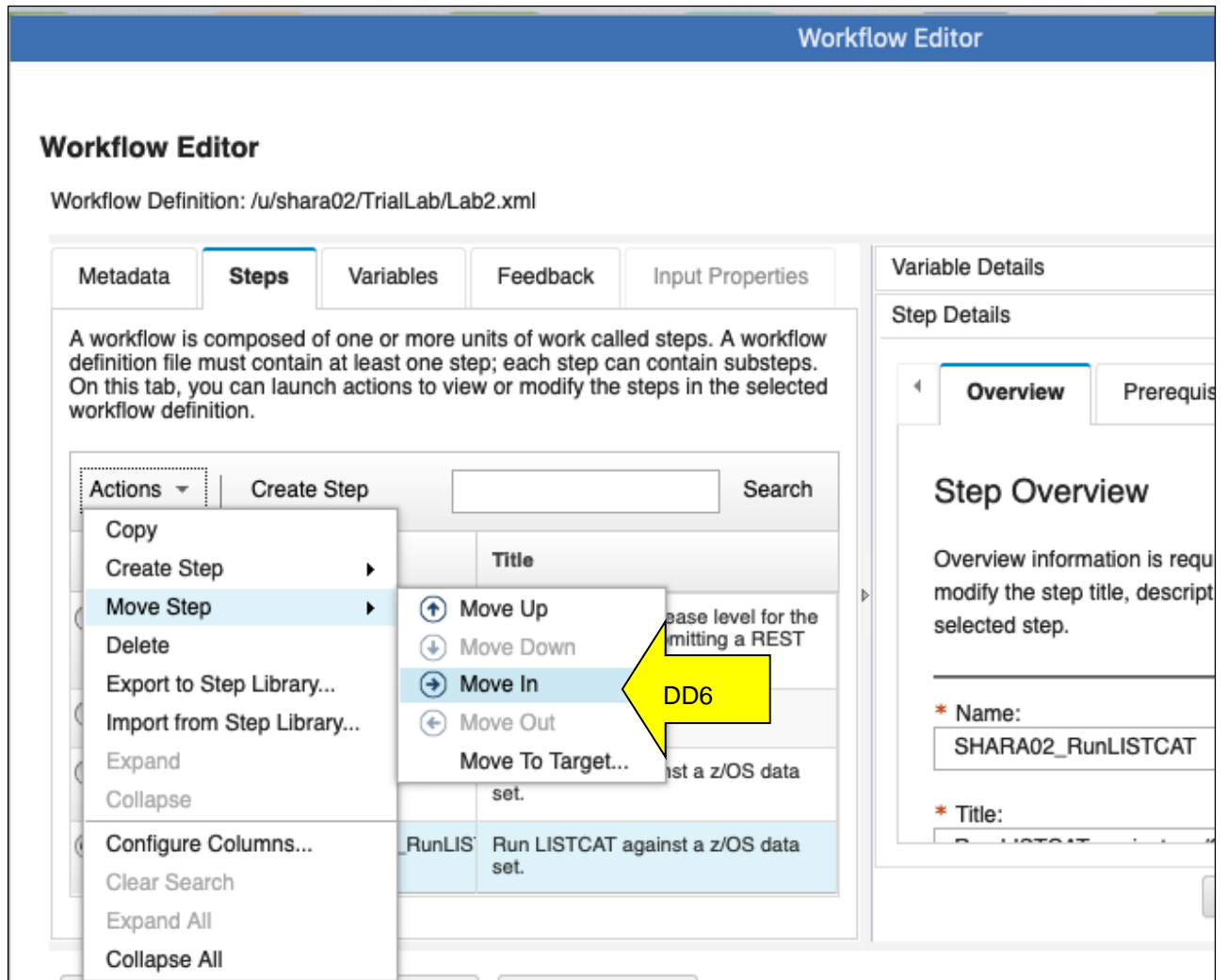
The Workflow Editor uses a process called *unmarshalling* to extract the contents of these files into its cache. Subsequently, when you save the edited file, the Workflow Editor uses a process called *marshalling* to create a single, consolidated workflow definition file that represents all of the requisite files. The resulting object is functionally equivalent to the original workflow structure, and the references to external files **are removed**.

During unmarshalling, there is a logical "break" in connection between the workflow definition and the external referenced XML files. The unmarshalling process creates a single workflow definition file as output.

When a save is done, there is an update to the cached version of the workflow definition file, and also a write out of the XML to the file that was opened initially. Variables or step XML files that were referenced from the opened definition are not written out. This is where you should be aware of a "break", as the changes are not written to a file template file.

Moving steps around

DD6: At this point, you have the step added after the parent step you selected. For this deep dive, let's pretend that we wanted this imported step to be prior parent step in our Workflow. It's easy to do, just click on the new step, **Actions**, then **Move Step >**, then **Move In**.



You can now see it's moved up into the parent step and is now Step 2.2 in the Workflow.

Workflow Editor

Workflow Editor

Workflow Definition: /u/shara02/TrialLab/Lab2.xml

Metadata

Steps

Variables

Feedback

Input Properties

A workflow is composed of one or more units of work called steps. A workflow definition file must contain at least one step; each step can contain substeps. On this tab, you can launch actions to view or modify the steps in the selected workflow definition.

Actions ▾

Create Step

Search

	Step No.	Name	Title
<input type="radio"/>	1	■ IBM-Toolbox-ObtainZosRelease	Obtain the z/OS release level for the local system by submitting a REST request
<input type="radio"/>	2	▢ RunStuff	Run some z/OS commands
<input type="radio"/>	2.1	■ RunLISTCAT	Run LISTCAT against a z/OS data set.
<input checked="" type="radio"/>	2.2	■ SHARA02_Run	Run set. z/OS data

Total: 4 Selected: 1

Variable Details

Step Details

Overview

Prerequisites

Instructions

Step Overview

Overview information is required for every step. On modify the step title, description, and other basic in selected step.

* Name:

SHARA02_RunLISTCAT

* Title:

Help

50

At the end of this exercise go ahead and delete the workflow you created.

Select the workflow you create. Then under **Actions** select **Delete**

The screenshot shows the IBM Workflows application window. The title bar is blue with the text "Workflows" and standard window controls. Below the title bar, there's a "Workflows" section with a description: "Simplifies tasks through guided step-based workflows, and provides administrative functions for assigning workflow responsibilities and tracking progress." To the right of this description is a "Search" input field. Below the description is a table of workflows. The table has columns: "Description", "Version", "Vendor", "Access", "Owner", and "System". A yellow arrow points to the "Delete" option in the "Actions" menu, which is open over the first row of the table. The "Delete" option is highlighted in blue. Other options in the menu include "View Properties", "Open", "Modify...", "Cancel", "Archive", "Stop Automation", "Generate Feedback Summary", "Create New Based on Existing", "Reactivate...", "Update Workflow Steps", "Create Workflow...", "Customize JOB Statement", "Workflows View", "Select All", "Deselect All", "Configure Columns...", and "Hide Filter Row".

Description	Version	Vendor	Access	Owner	System
Container Extensions	1.0.17	IBM	Public	danjast	SHARPLEX.S2
Customization for the zOSMF plug-ins	2.2.2(PI59506)	IBM	Public	danjast	SHARPLEX.S2
Configure a IBM zOS Container Extensions Appliance Instance.	1.0.3	IBM	Public	danjast	SHARPLEX.S2
Workflow for DAN	1	IBM	Public	danjast	SHARPLEX.S2
S V2R4 Upgrade Workflow from zOS R3	2.0	IBM	Public	shara01	SHARPLEX.S2
Workflow using the zOSMF Workflow Editor	1.0	SHARA02 Corporation Inc.	Public	shara02	SHARPLEX.S2

Appendix of sample provided for this lab

Here is the sample of the **listcat** JCL. Notice all the **\$instance-variable** locations for the Velocity engine.

```
VIEW          /u/mwalle/zosmf/labs/listcat          Columns 00001 00072
Command ==>                                         Scroll ==> CSR
***** ***** Top of Data *****
000001 //LISTCAT EXEC PGM=IDCAMS
000002 //SYSPRINT DD SYSOUT=*
000003 //SYSIN    DD *
000004 LISTCAT ALL ENT ($instance-DSN)
***** ***** Bottom of Data *****
```