z/OSMF labs: A smorgasbord of goodies to try out!

Create your own Workflow (Workflow Editor)

TechU

2019 IBM Systems Technical University

IBM

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).

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

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. Create steps in your Workflow.

__ 3. Define variables to be used in your Workflow.

__ 4. Provide defaults and variable specifications.

__ 5. Save your steps into a library for others to use.

__ 6. Retrieve steps from a library that others have provided.

__ 7. Save your Workflow.

__ 8. Try your Workflow out to make sure it is as you desire.

__ 9. Edit an existing Workflow to add, remove, and change steps.

__ 10. 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. Allocate a traditional (or "legacy") z/OS sequential data set.
2. Copy that sequential data set into a directory in a z/OS UNIX file system.

Each of these steps will be done using JCL. The Workflow user can specify some specific values to use.

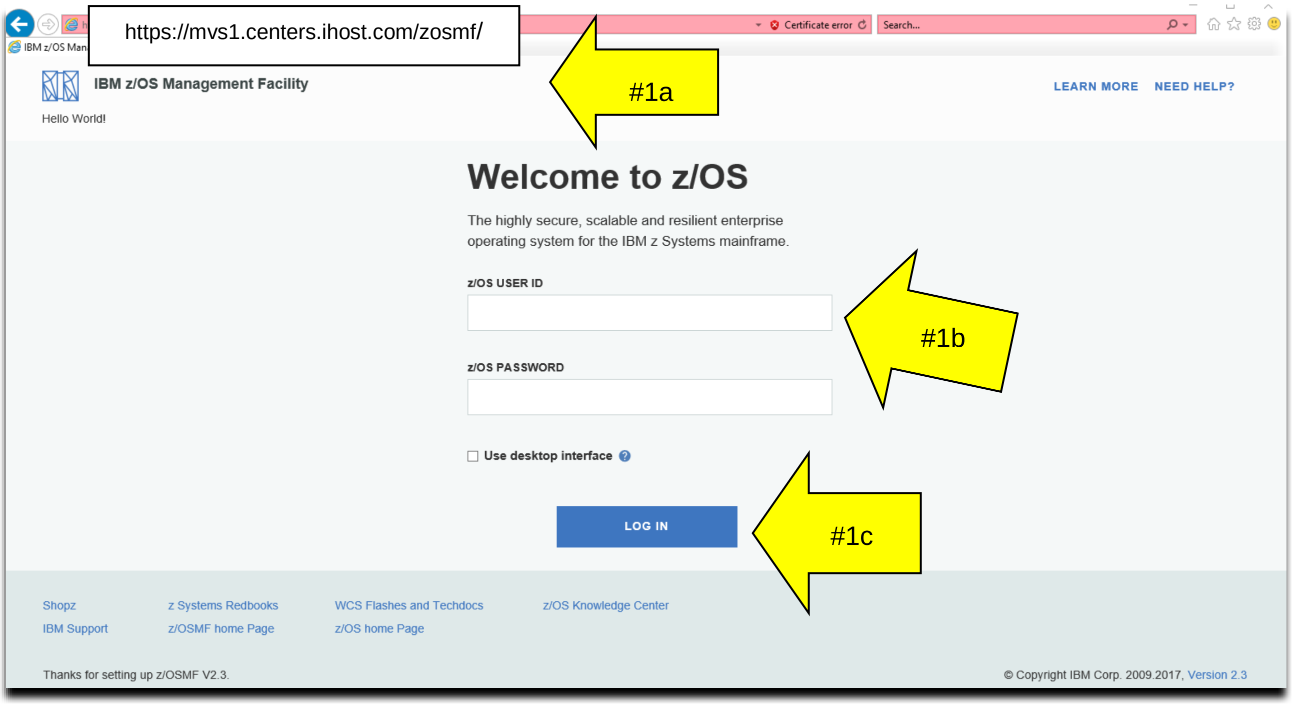
Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

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 V2.3 system.

a. Go to https://mvs1.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 not some of the samples we use you will need to supply yourself, using the Appendix to find those samples.) Click on "Log in".

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.



Welcome to z/OS

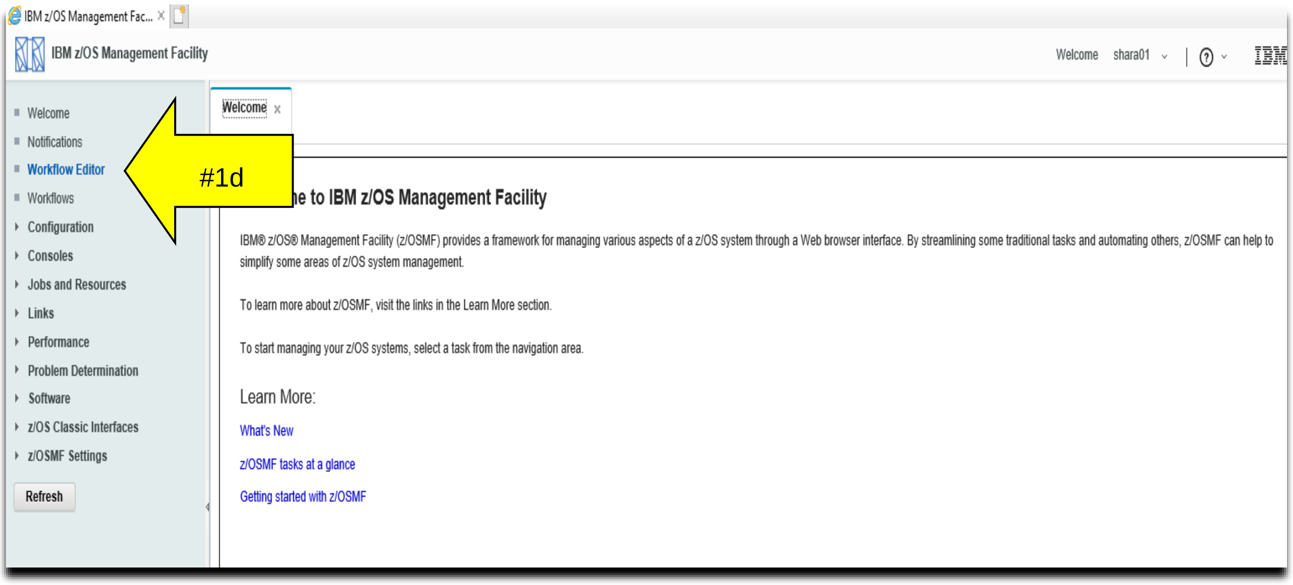
The highly secure, scalable and resilient enterprise operating system for the IBM z Systems mainframe.

Thanks for setting up z/OSMF V2.3.

© Copyright IBM Corp. 2009.2017, Version 2.3

Enter into the Workflow Editor function.

c. Click on “Workflow Editor”.



...me to IBM z/OS Management Facility

IBM® z/OS® Management Facility (z/OSMF) provides a framework for managing various aspects of a z/OS system through a Web browser interface. By streamlining some traditional tasks and automating others, z/OSMF can help to simplify some areas of z/OS system management.

To learn more about z/OSMF, visit the links in the Learn More section.

To start managing your z/OS systems, select a task from the navigation area.

Learn More:

What's New

z/OSMF tasks at a glance

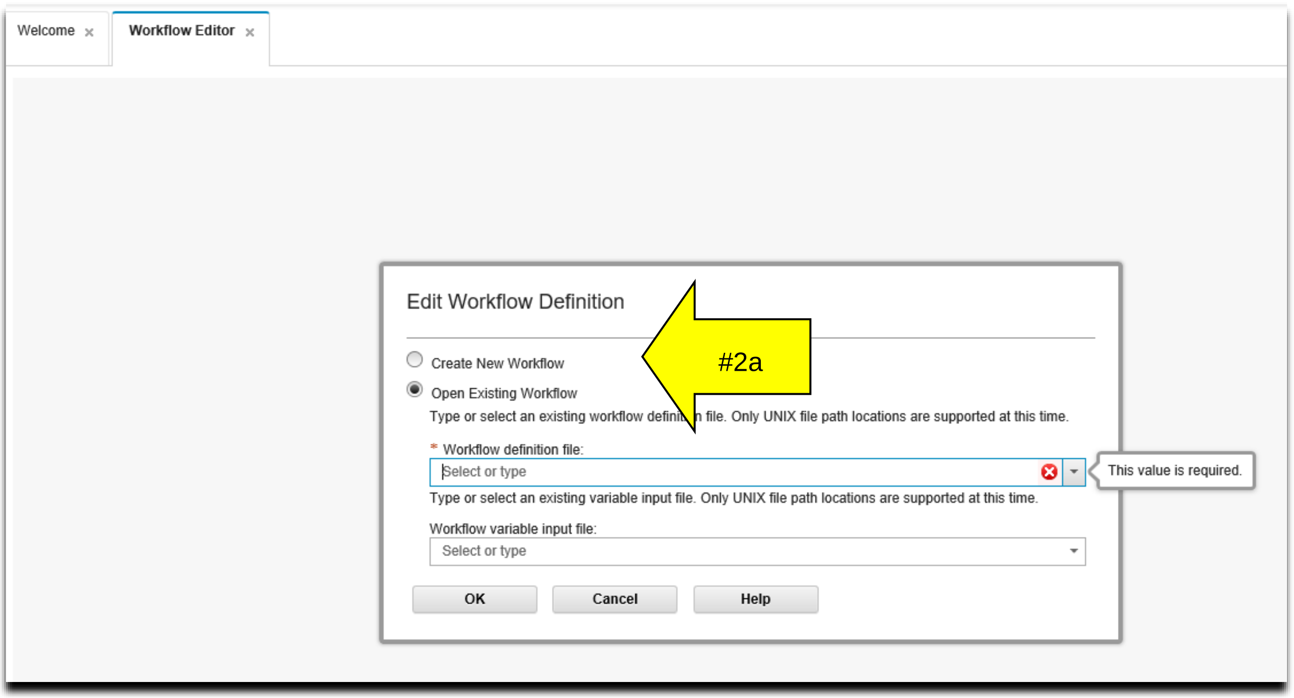
Getting started with z/OSMF

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

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.



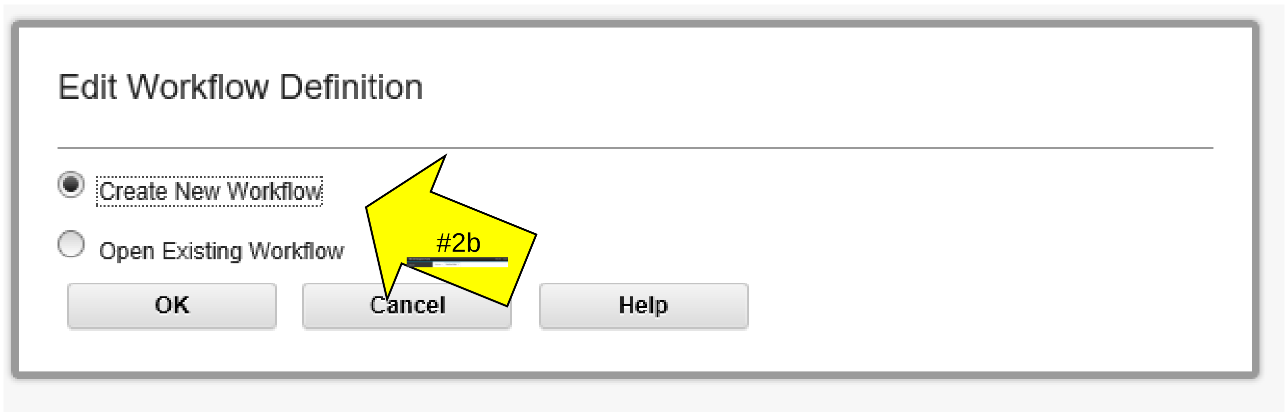
Edit Workflow Definition

○ Create New Workflow
● Open Existing Workflow
Type or select an existing workflow definition file. Only UNIX file path locations are supported at this time.

* Workflow definition file:
Select or type
This value is required.
Type or select an existing variable input file. Only UNIX file path locations are supported at this time.

Workflow variable input file:
Select or type

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



Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

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.

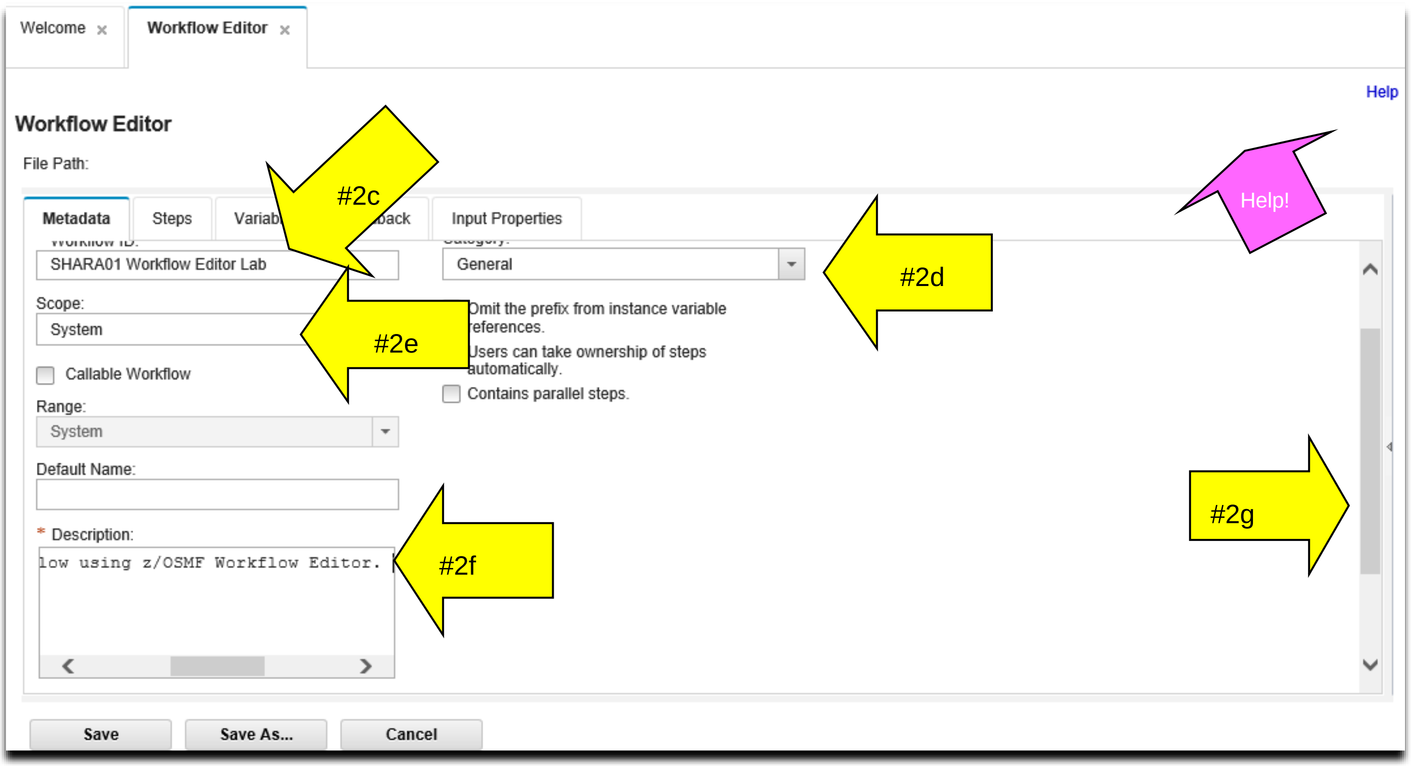
- A suggested name: ***YourUserID Workflow Editor Lab* ,** where ***YourUserID*** is your assigned userid, such as SHARB15.

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.

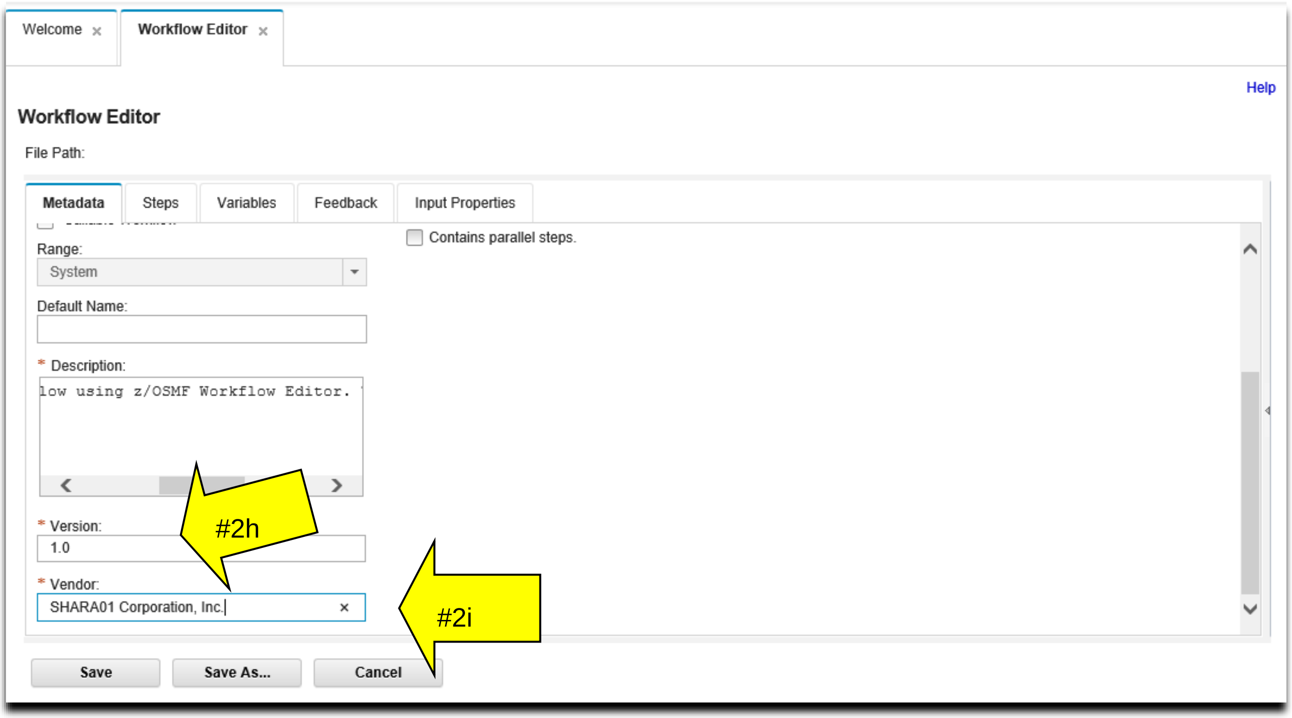


Workflow ID: SHARA01 Workflow Editor Lab
Category: General
Scope: System
☐ Omit the prefix from instance variable references.
☐ Users can take ownership of steps automatically.
☐ Callable Workflow
☐ Contains parallel steps.
Range: System
Default Name:
* Description: low using z/OSMF Workflow Editor.

h. Type in something you like for **Version**, such as 1.0. Think of this as the SMP/E REWORK value, if you like, and you are familiar with SMP/E.

i. Type in something you like for **Vendor.** Who designed this Workflow? What company created this Workflow?

Do not click on "Save" at this time. This lab will take you through each tab.



Workflow Editor

File Path:

Contains parallel steps.

Range: System

Default Name:

* Description: low using z/OSMF Workflow Editor.

* Version: 1.0

* Vendor: SHARA01 Corporation, Inc.

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

3. Steps for your Workflow.

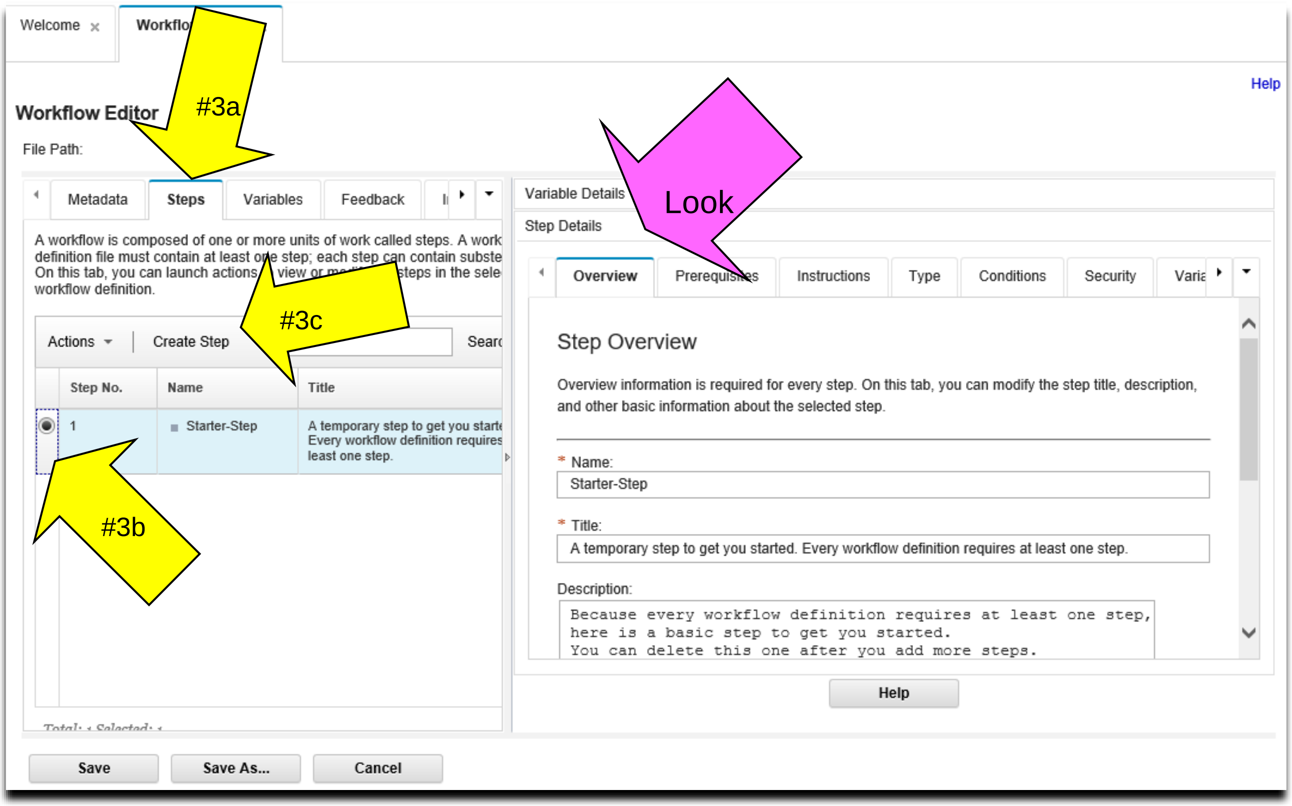
Now, we'll add some simple steps to our new Workflow.

a. 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.

b. 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 actually 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.

c. Click on **Create Step** to add the first "real" step to the new Workflow.



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

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

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: Starter-Step

* Title: A temporary step to get you started. Every workflow definition requires at least one step.

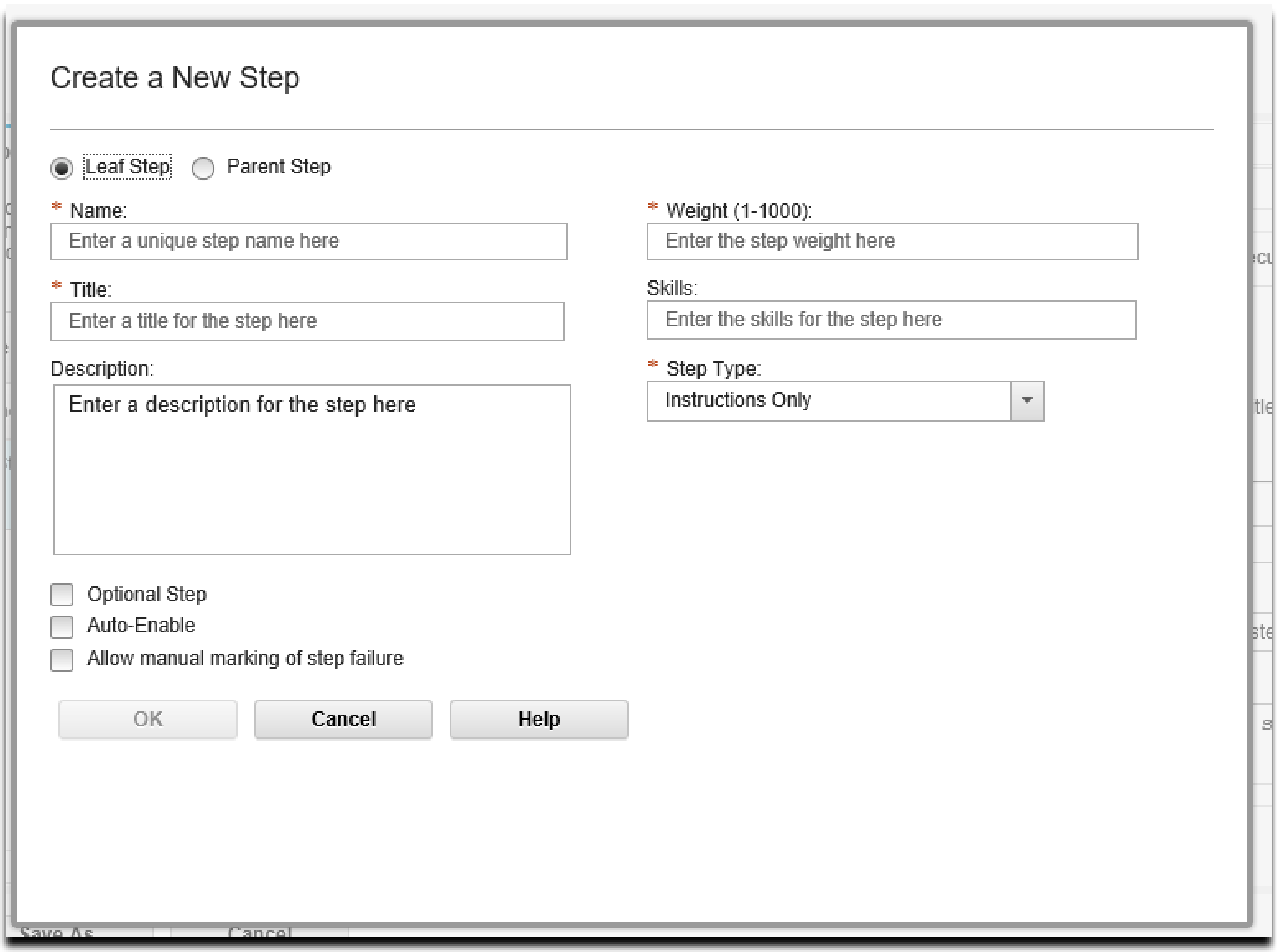
Description:

```
Because every workflow definition requires at least one step,
here is a basic step to get you started.
You can delete this one after you add more steps.
```

Total: 1 Selected: 1

d. 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

* Title: Enter a title for the step here

Description: Enter a description for the step here

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

Skills: Enter the skills for the step here

* Step Type: Instructions Only

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

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

d. 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 ***DataStepCreateAndCopy.***

- 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.

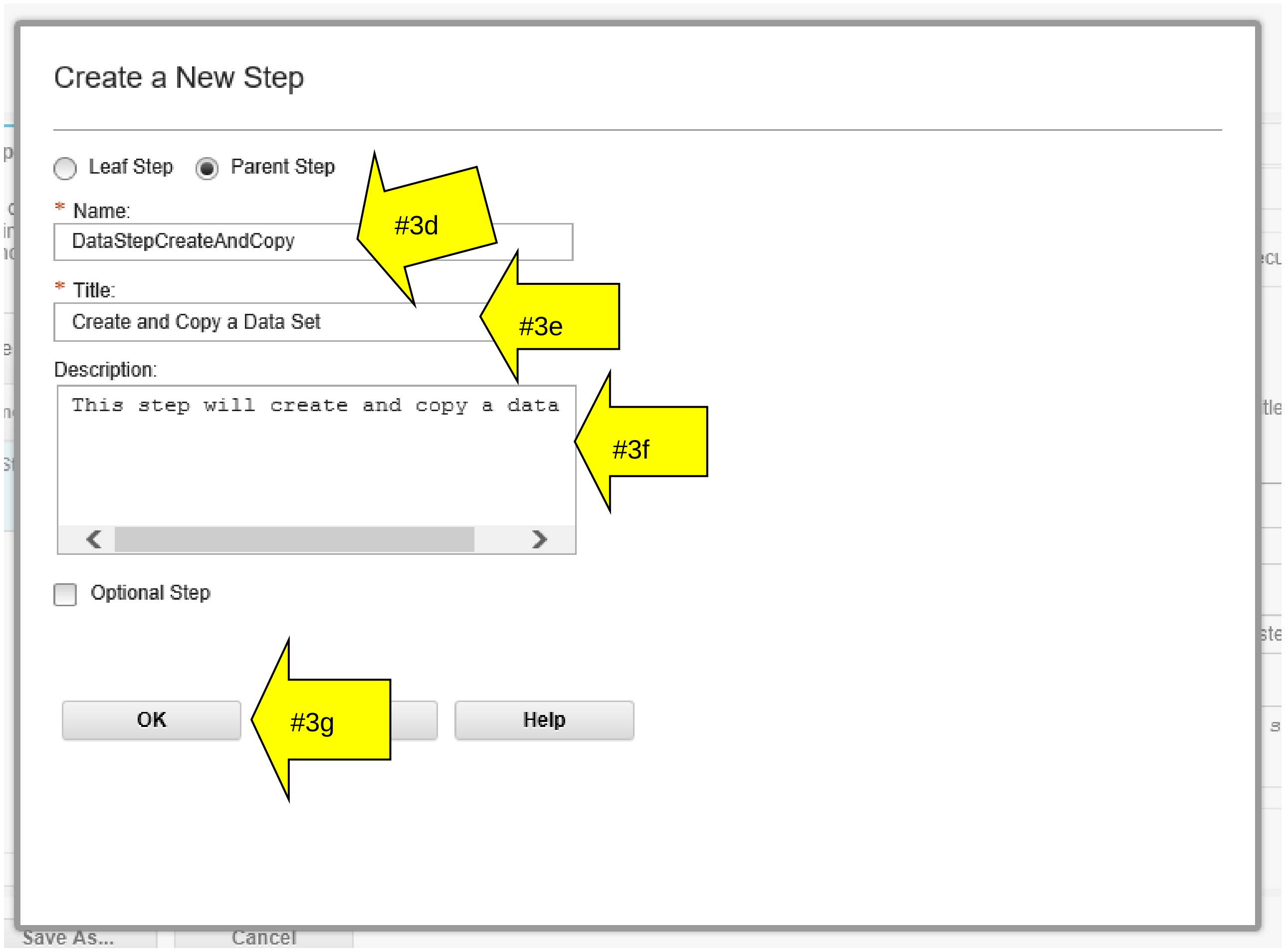
e. Provide a **Title.** Again, for simplicity and for matching these lab instructions call it ***Create and Copy a Data Set.***

- 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.

f. Provide a **Description.** Type whatever you want here.

- You can have a long description that is up to 500 characters.

g. Click on **OK.** This will add a parent step to your new Workflow under which you can now add Leaf Steps.



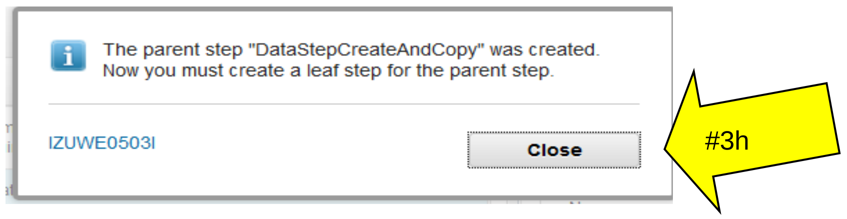
Leaf Step [✓ Parent Step]

* Name: DataStepCreateAndCopy

* Title: Create and Copy a Data Set

Description: This step will create and copy a data

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



The parent step "DataStepCreateAndCopy" was created.
Now you must create a leaf step for the parent step.

IZUWE0503I

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

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 presentsthe **Leaf Step** dialog to you.

i. Keep the **Leaf Step** radio button selected this time.

j. 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 ***AllocDSN.***

   - 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.

k. Provide a **Title.** Again, for simplicity and for matching these lab instructions call it ***Allocate a traditional 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.

l. Provide a **Description.** Type whatever you want here.

   - You can have a long description that is up to 500 characters.

m. 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.

n. 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.

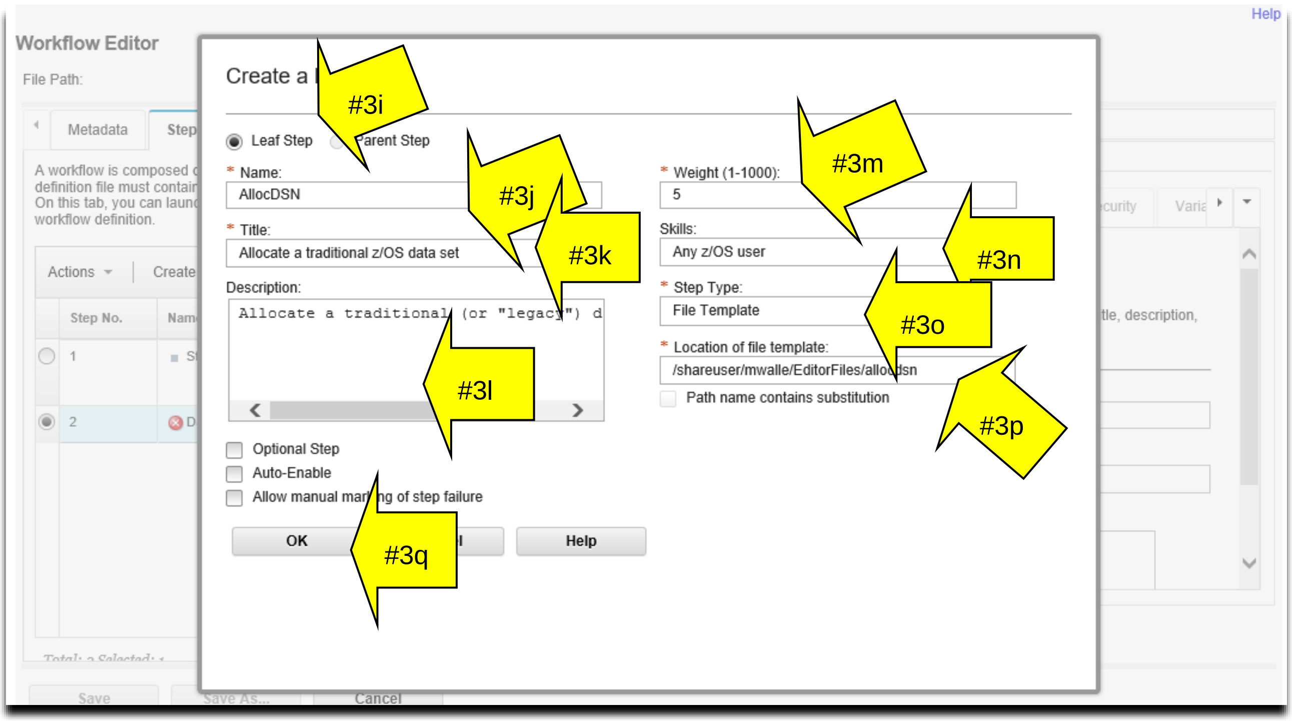
o. Indicate a **Step Type** of **File Template.**

   - 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.

p. When you have a **File Template,** you need to say where it is.

   - Type in ***/shareuser/mwalle/EditorFiles/allocdsn*** 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.

q. Click on **OK.** This will add a leaf step under the parent step you created.



Create a Leaf Step

● Leaf Step ○ Parent Step

* Name: AllocDSN

* Title: Allocate a traditional z/OS data set

Description: Allocate a traditional (or "legacy") d

* Weight (1-1000): 5

Skills: Any z/OS user

* Step Type: File Template

* Location of file template: /shareuser/mwalle/EditorFiles/allocdsn

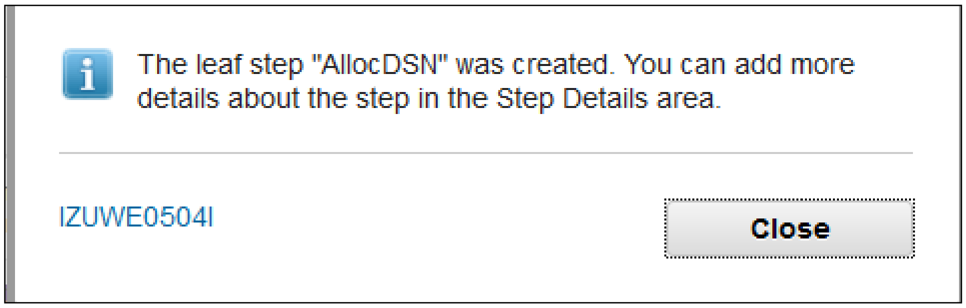
☐ Path name contains substitution

☐ Optional Step

☐ Auto-Enable

☐ Allow manual marking of step failure

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



The leaf step "AllocDSN" was created. You can add more details about the step in the Step Details area.

IZUWE0504I

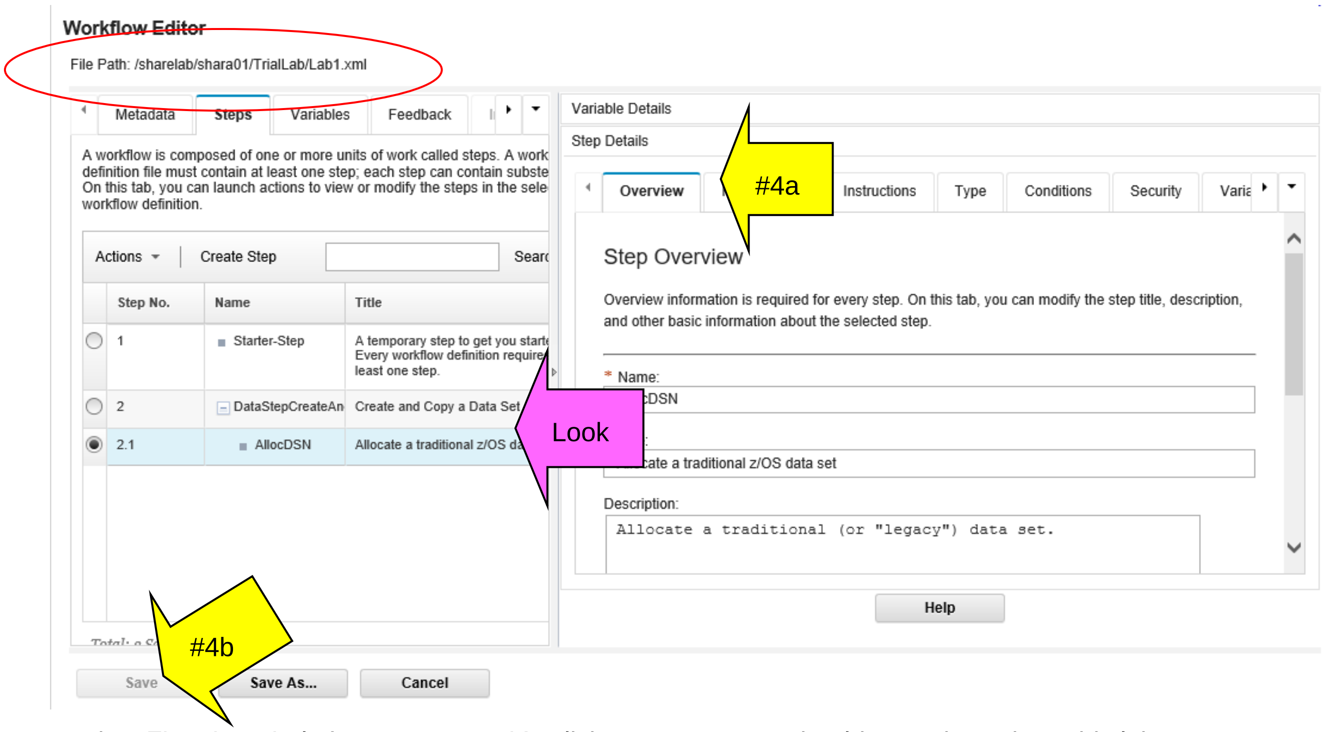
A wonderful thing to know! The Workflow Editor will never allow you to produce an invalid Workflow. In other words, if you use the Editor you can be assured that a user can create a Workflow that you produced. It might not be what you had intended to create, but that is a different story ☺.

Materials may not be reproduced in whole or in part without the prior written permission of IBM.

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.



Workflow Editor

File Path: /sharelab/shara01/TrialLab/Lab1.xml

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

	Step No.	Name	Title
○	1	Starter-Step	A temporary step to get you starte Every workflow definition require least one step.
○	2	DataStepCreateAn	Create and Copy a Data Set
●	2.1	AllocDSN	Allocate a traditional z/OS da

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: AllocDSN

Title: Allocate a traditional z/OS data set

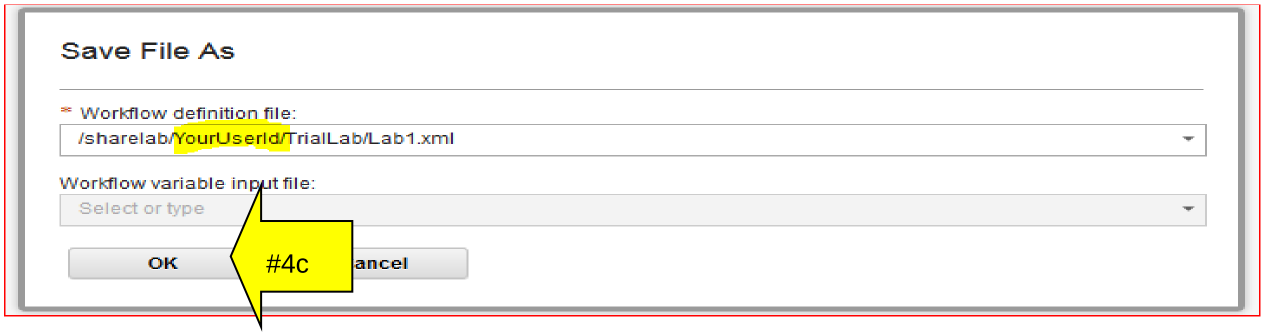
Description:

```
Allocate a traditional (or "legacy") data set.
```

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 ***/sharelab/YourUserId/TrialLab/Lab1.xml*** as the location, where ***YourUserId*** is your assigned userid. Use all lower case for ***YourUserId*** , such as ***sharea01.*** Remember, pathnames are case sensitive.



Save File As

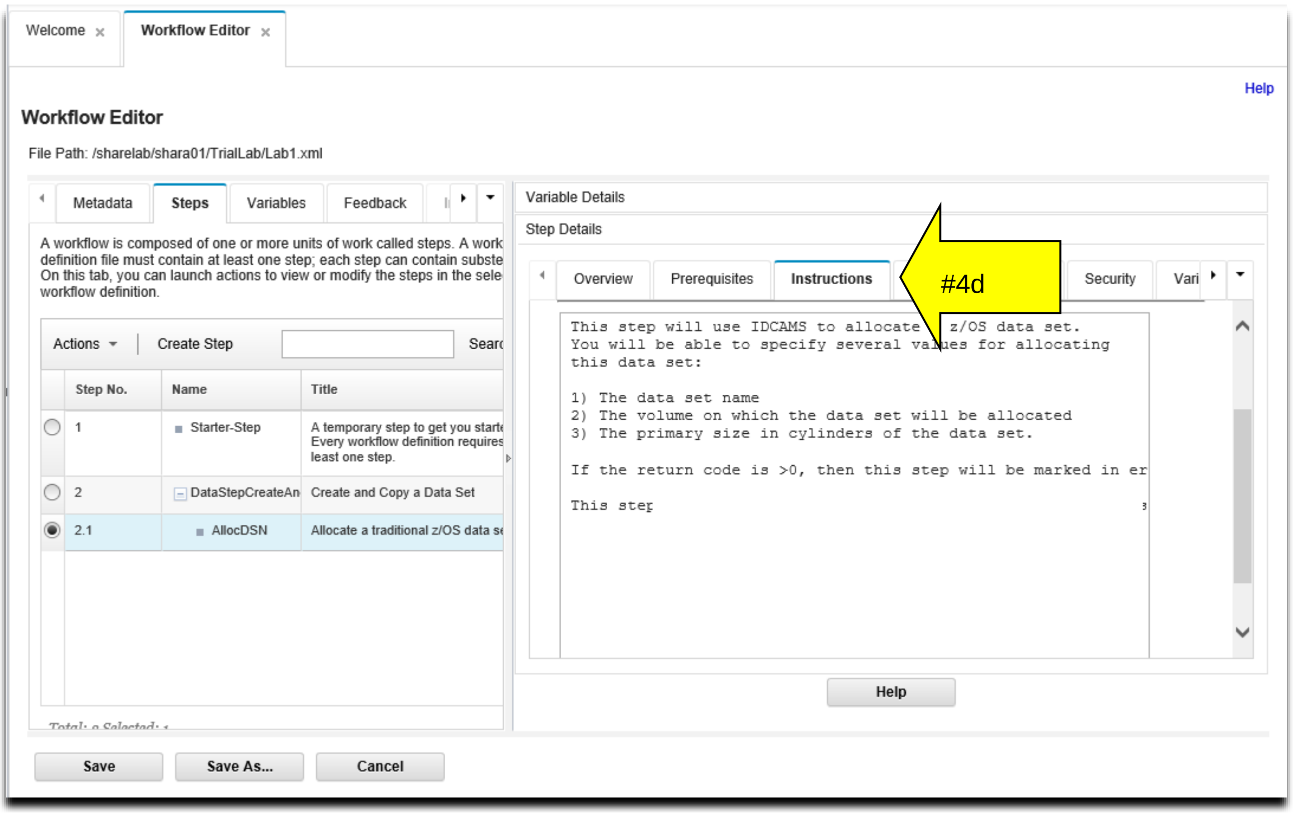
* Workflow definition file: /sharelab/YourUserId/TrialLab/Lab1.xml

Workflow variable input file: Select or type

Let's create some more **Step Details** for our ***AllocDSN*** 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 he is performing this step. Note that in the instructions provided below for our sample Workflow, we are telling the user what input he should be prepared to answer.

- *What are good instructions?* Here, a user will want to know how to perform this step. For instancein our Workflow we are creating, the user will have to be prepared to know what this step does and the answer to two variables (the data set name, and the volume). This is the type of helpful information that good instructions should contain.



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

Step No.	Name	Title
1	Starter-Step	A temporary step to get you starte Every workflow definition requires least one step.
2	DataStepCreateAn	Create and Copy a Data Set
2.1	AllocDSN	Allocate a traditional z/OS data se

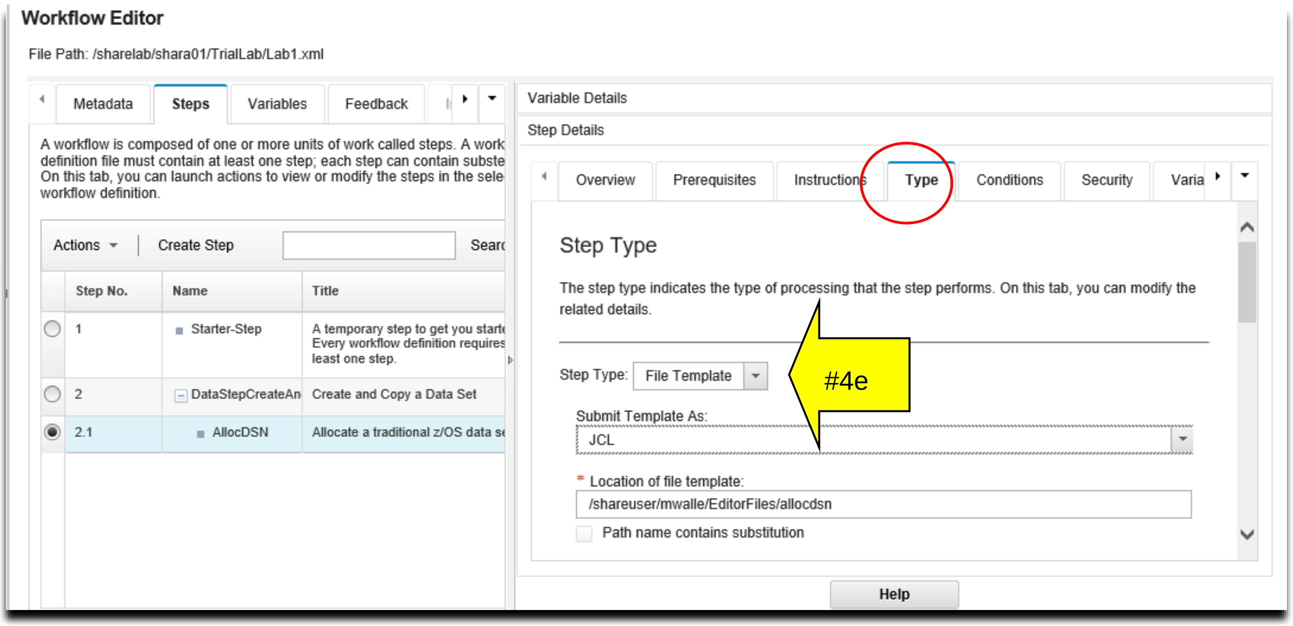
```
This step will use IDCAMS to allocate   z/OS data set.
You will be able to specify several values for allocating
this data set:

1) The data set name
2) The volume on which the data set will be allocated
3) The primary size in cylinders of the data set.

If the return code is >0, then this step will be marked in er

This step
```

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Next up under **Step Details** for our ***AllocDSN*** 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.



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

Step No.	Name	Title
1	Starter-Step	A temporary step to get you starte Every workflow definition requires least one step.
2	DataStepCreateAn	Create and Copy a Data Set
2.1	AllocDSN	Allocate a traditional z/OS data se

Step Type

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

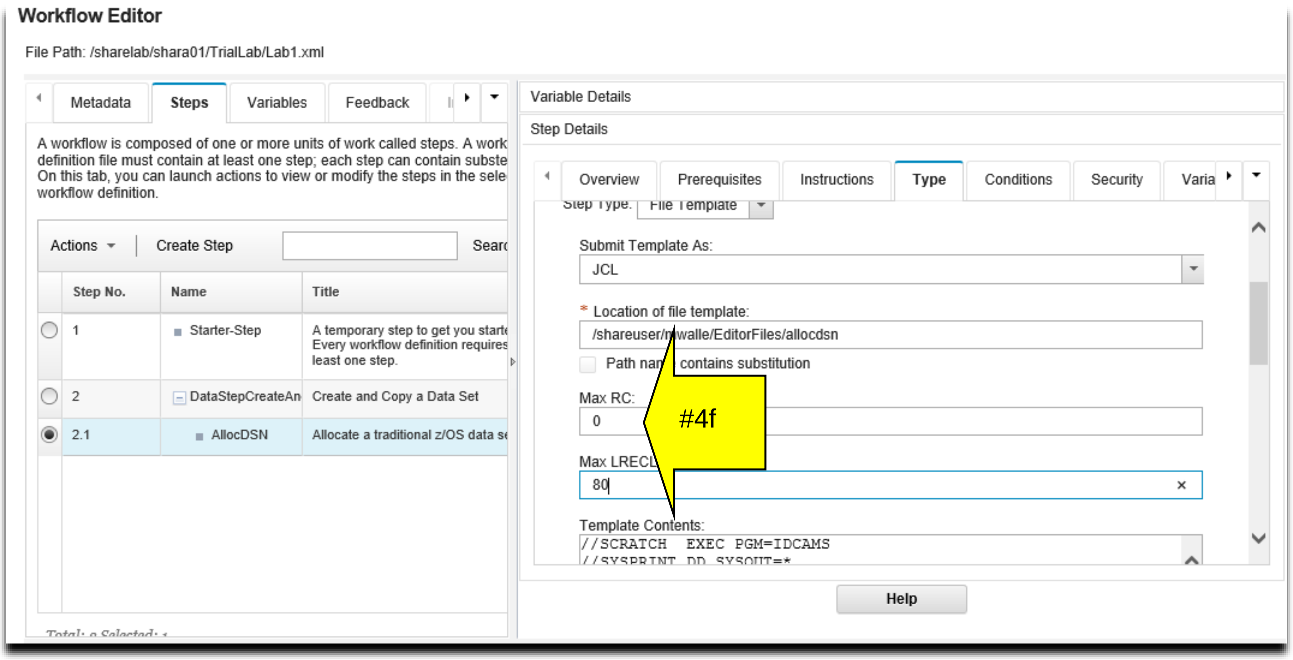
Step Type: File Template

Submit Template As: JCL

* Location of file template: /shareuser/mwalle/EditorFiles/allocdsn

Path name contains substitution

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.**



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

Step No.	Name	Title
1	Starter-Step	A temporary step to get you starte Every workflow definition requires least one step.
2	DataStepCreateAn	Create and Copy a Data Set
2.1	AllocDSN	Allocate a traditional z/OS data se

Step Type: File Template

Submit Template As: JCL

* Location of file template: /shareuser/mwalle/EditorFiles/allocdsn

Path name contains substitution

Max RC: 0

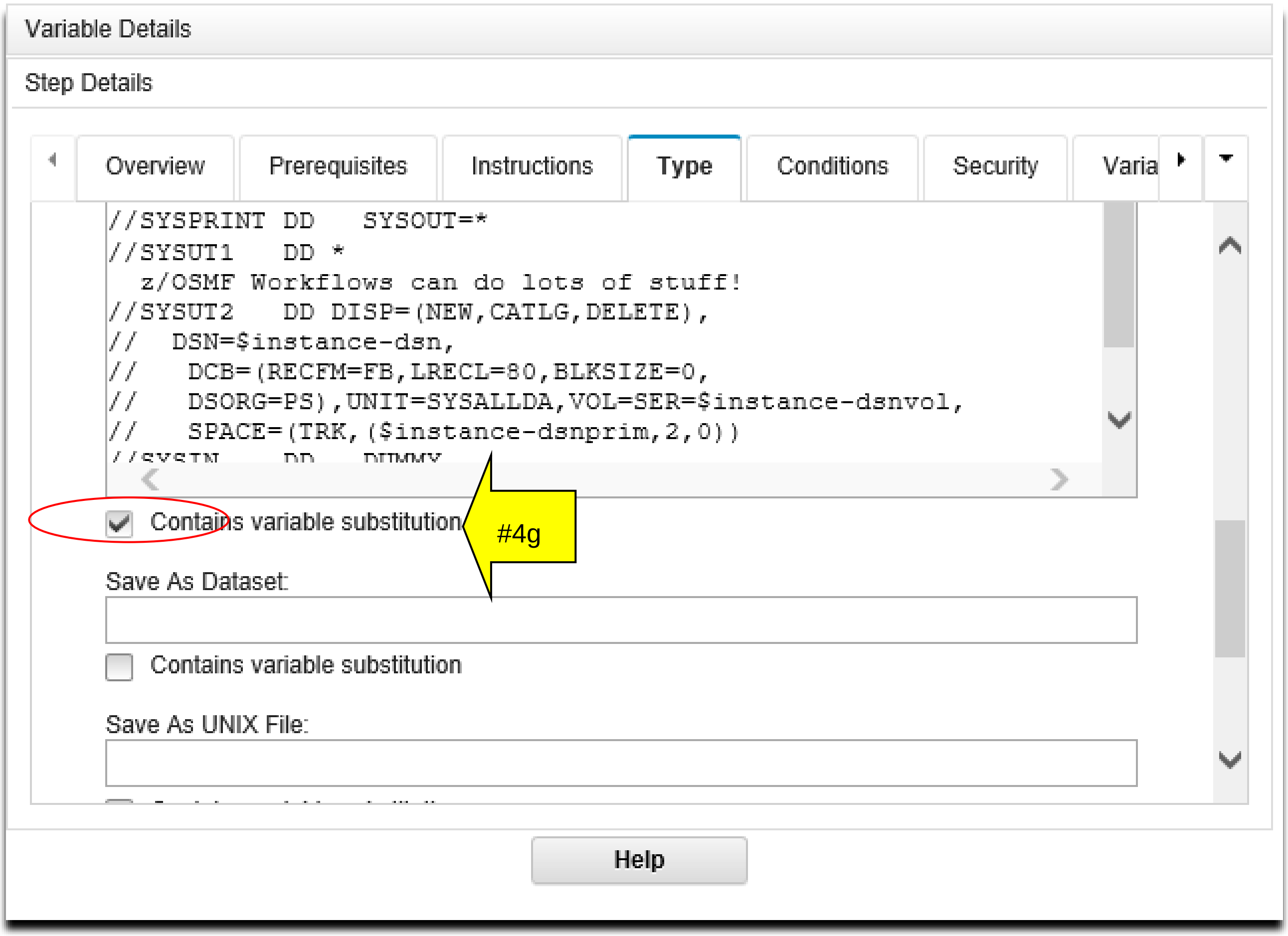
Max LRECL: 80

Template Contents:

```
//SCRATCH  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
```

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.



```
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD *
  z/OSMF Workflows can do lots of stuff!
//SYSUT2   DD DISP=(NEW,CATLG,DELETE),
//  DSN=$instance-dsn,
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,
//   DSORG=PS),UNIT=SYSALLDA,VOL=SER=$instance-dsnvol,
//   SPACE=(TRK,($instance-dsnprim,2,0))
```

☑ Contains variable substitution

Save As Dataset:

☐ Contains variable substitution

Save As UNIX File:

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

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

5. Variables in your Workflow.

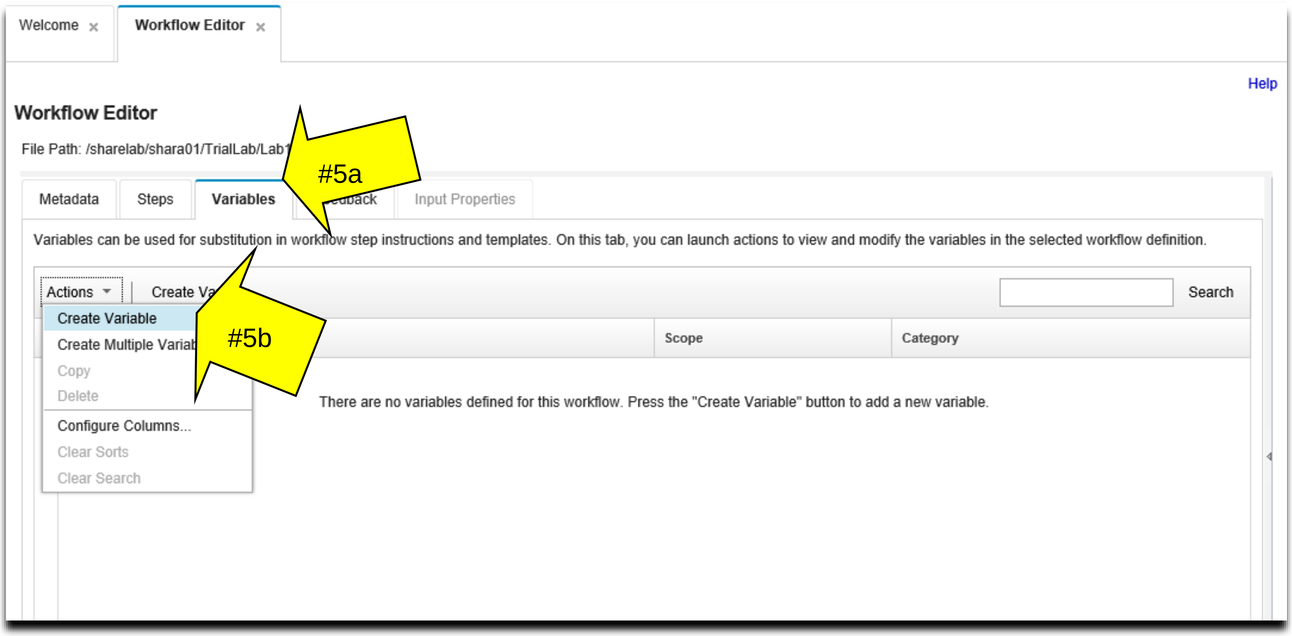
We have variables in our Workflow steps, if you looked closely in the JCL template. That is how you bring value to Workflow steps and allow the user to give you some input. You put that input into a variable and then substitute it in the template when it is time to run the JCL and is then submitted by 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'll not explore Input Properties in this lab.

Let's create variables in the Editor:

a. Click on the **Variables** tab.

b. Then, click on **Actions** drop down, and select **Create Variable.** Alternatively, you could also click on **Create Variable** right next to **Actions**.



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.

There are no variables defined for this workflow. Press the "Create Variable" button to add a new variable.

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 ***New data set name to allocate.*** 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.

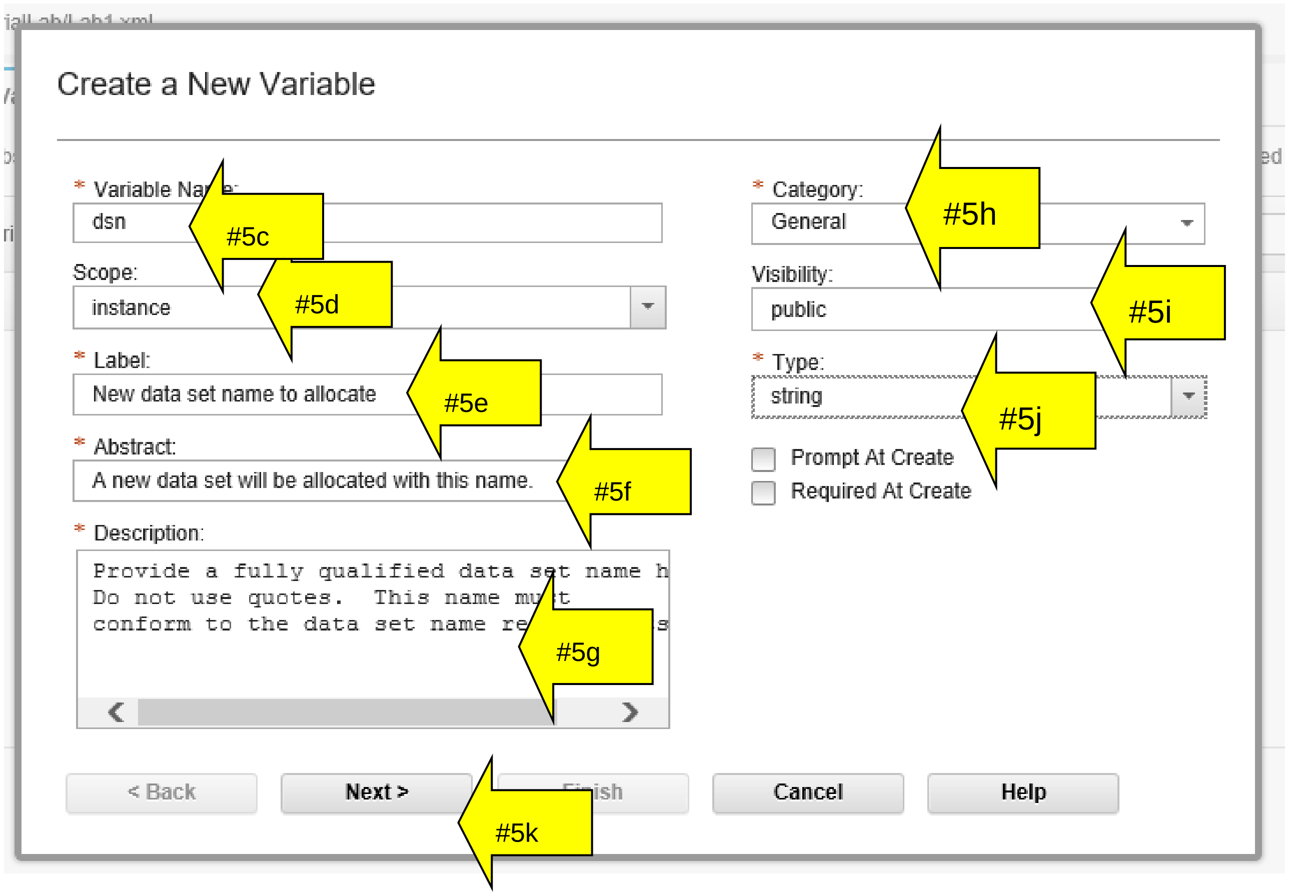
g. For **Description** type a longer description of what the user might need to know about this variable.

h. For **Category** select **General.** This is the only option at this point for **Category.**

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>.**


Create a New Variable

* Variable Name: dsn
Scope: instance
* Label: New data set name to allocate
* Abstract: A new data set will be allocated with this name.
* Description:
```
Provide a fully qualified data set name h
Do not use quotes.  This name mu t
conform to the data set name re         s
```
* Category: General
Visibility: public
* Type: string
Prompt At Create
Required At Create

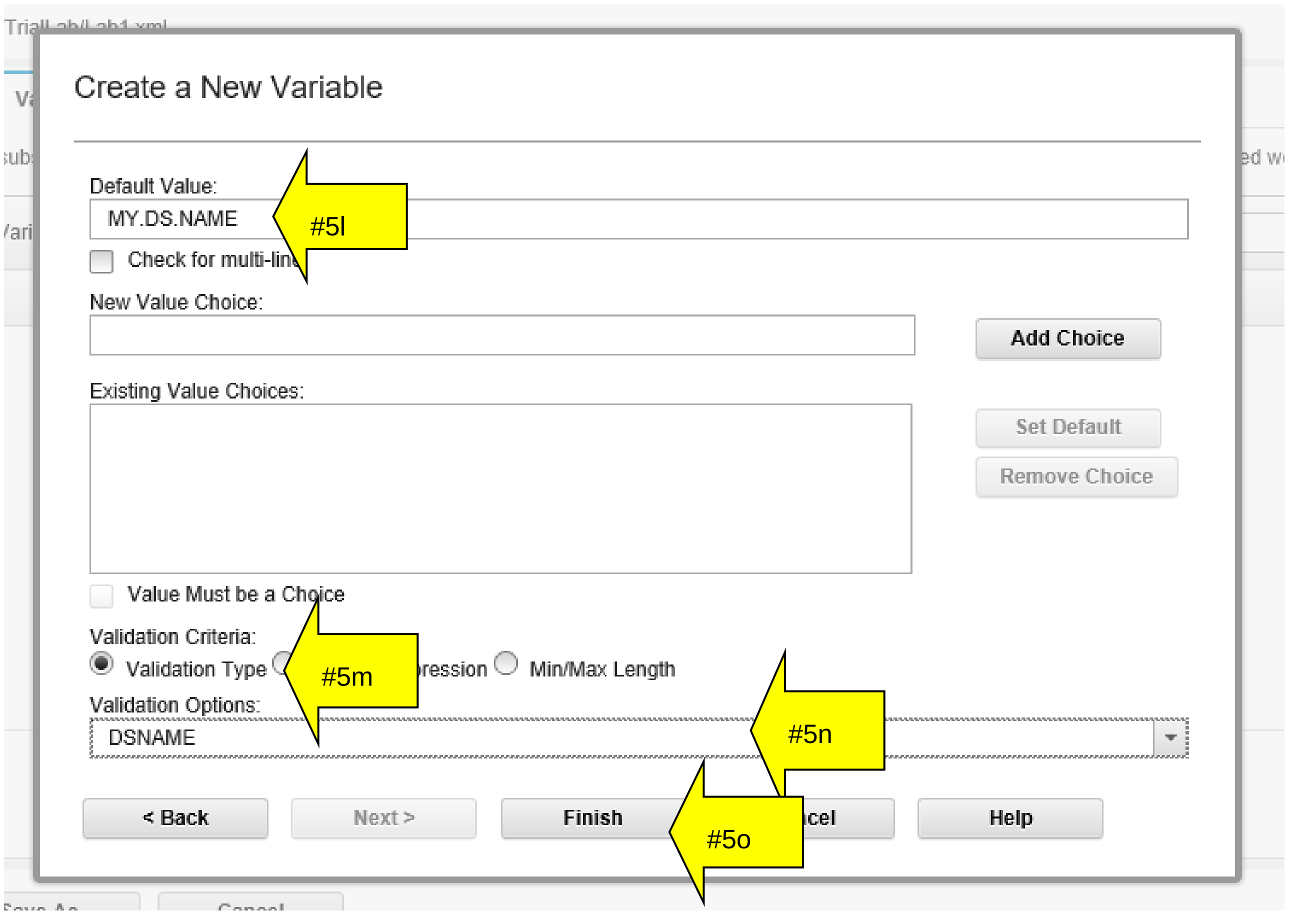
Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

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.**

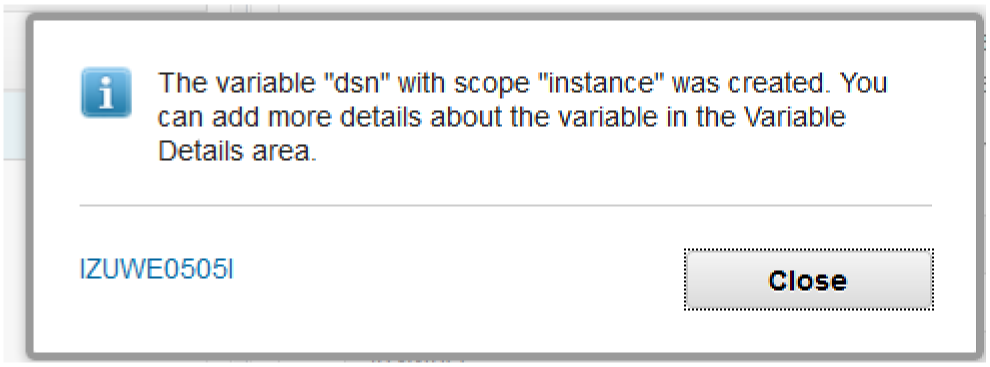

Create a New Variable

Default Value: MY.DS.NAME

Validation Criteria: Validation Type

Validation Options: DSNAME

You now have your first Workflow variable:


The variable "dsn" with scope "instance" was created. You can add more details about the variable in the Variable Details area.

IZUWE0505I

p. You will now create second and third variables all by yourself! The first variable will be used as the volume serial (***dsnvol***) during the allocation. The second variable will be used for the primary space allocation (***dsnprim***) during the allocation, in tracks. Use these values:

=> Use these values for the first variable:

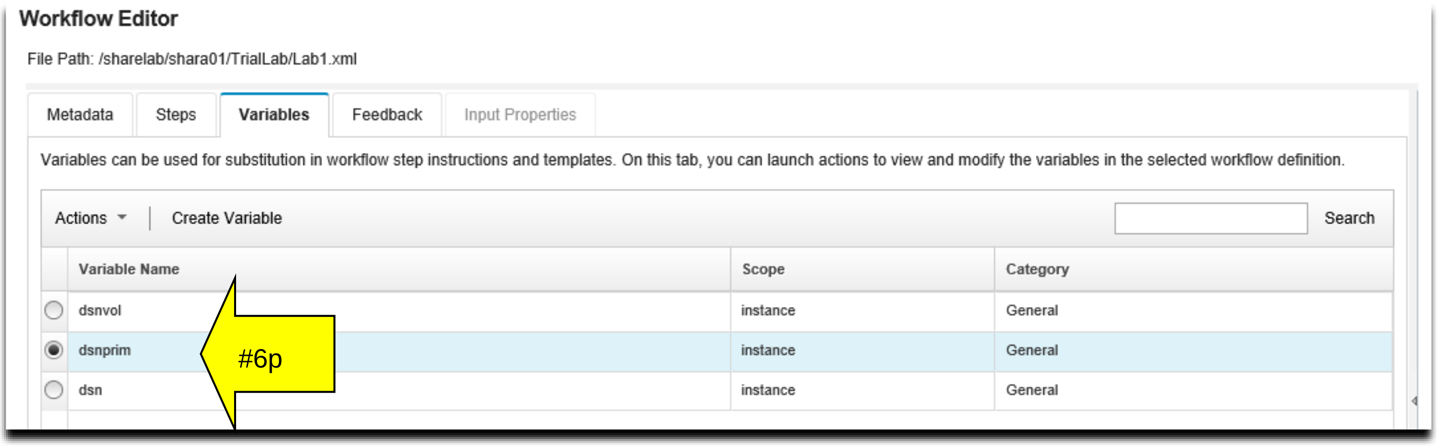
- **Variable Name** type ***dsnvol.***
- For Scope, select **Instance*.***
- For **Label** type ***Volume serial (VOL=SER=) for the new data set.***
- For **Abstract** type a short message that the user will see when filling in this variable in the Workflow.
- For **Description** type a long description of what the user might need to know about this variable.
- For **Category** select **General.**
- For **Visibility** select **public.** (Just to make it different from the variable ***dsn.***)
- For **Type** select **string.**
- For **Default Value** make it **VVVVVV**.
- Under **Validation Criteria,** select **Validation Type.**
- Under **Validation Option,** select **VOLSER.**

⇨ Use these values for the second variable:

- **Variable Name** type ***dsnprim.***
- For Scope, select **Instance*.***
- For **Label** type ***Primary data set allocation (in tracks) for the new data set.***
- For **Abstract** type a short message that the user will see when filling in this variable in the Workflow.
- For **Description** type a long description of what the user might need to know about this variable.
- For **Category** select **General.**
- For **Visibility** select **public.** (Just to make it different from the variable ***dsn.***)
- For **Type** select **integer.**
- For **Minimum value** make it **1**.
- For **Maximum value** make it **10.** (We don't have that much space on this system for labs!)
- For **Default Value** make it **2**.

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

When you are done, you should see three variables (***dsn*** , ***dsnvo,*** *and* ***dsnpriml***) under the **Variables** tab.



Workflow Editor

File Path: /sharelab/shara01/TrialLab/Lab1.xml

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.

	Variable Name	Scope	Category
○	dsnvol	instance	General
●	dsnprim	instance	General
○	dsn	instance	General

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:

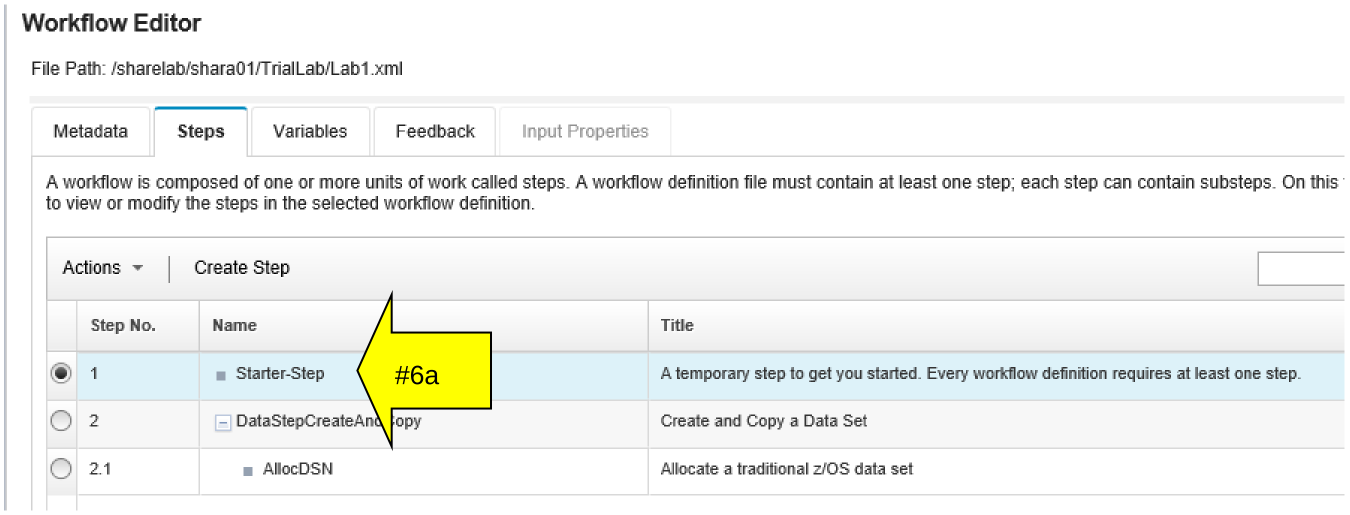
MY.DS.NAME (the default) or **MWALLE.LAB1.DSN** as appropriate.

6. Putting Variables in your Steps.

At this point, we have added one step (really one parent and one leaf step) and three variables in our Workflow. However, we don't have the variables *associated with that step*. 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.

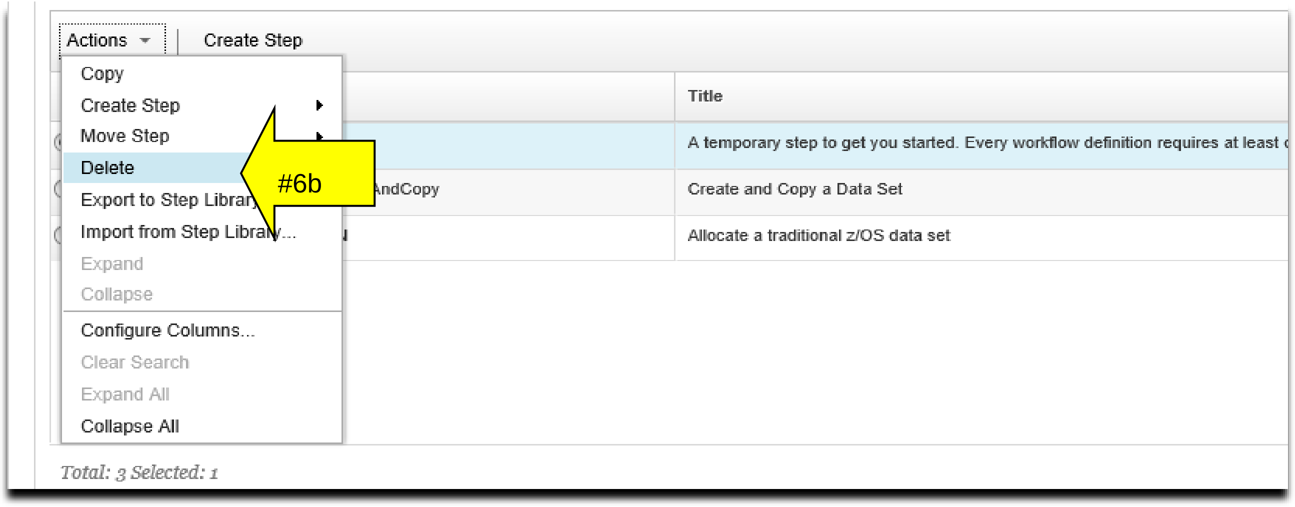

Workflow Editor

File Path: /sharelab/shara01/TrialLab/Lab1.xml

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 to view or modify the steps in the selected workflow definition.

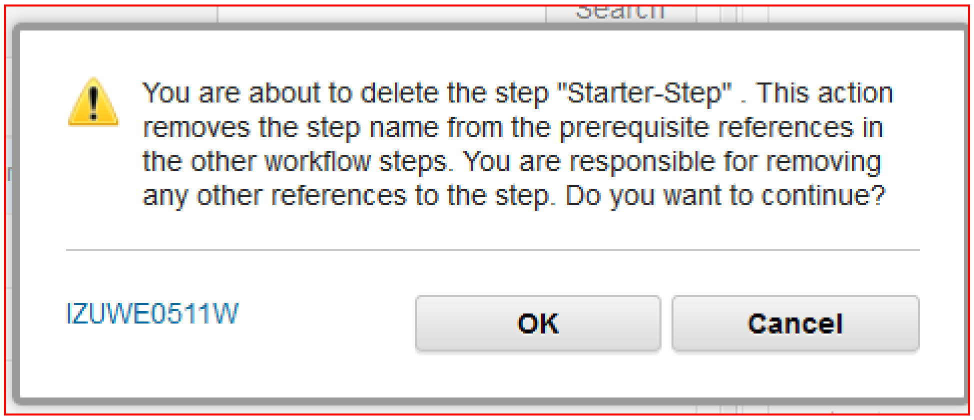
	Step No.	Name	Title
●	1	Starter-Step	A temporary step to get you started. Every workflow definition requires at least one step.
○	2	DataStepCreateAndCopy	Create and Copy a Data Set
○	2.1	AllocDSN	Allocate a traditional z/OS data set

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


Actions
Create Step
Copy
Create Step
Move Step
Delete
Export to Step Library...
Import from Step Library...
Expand
Collapse
Configure Columns...
Clear Search
Expand All
Collapse All
#6b
Title
A temporary step to get you started. Every workflow definition requires at least
Create and Copy a Data Set
Allocate a traditional z/OS data set
Total: 3 Selected: 1

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

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.



You are about to delete the step "Starter-Step" . This action removes the step name from the prerequisite references in the other workflow steps. You are responsible for removing any other references to the step. Do you want to continue?

IZUWE0511W

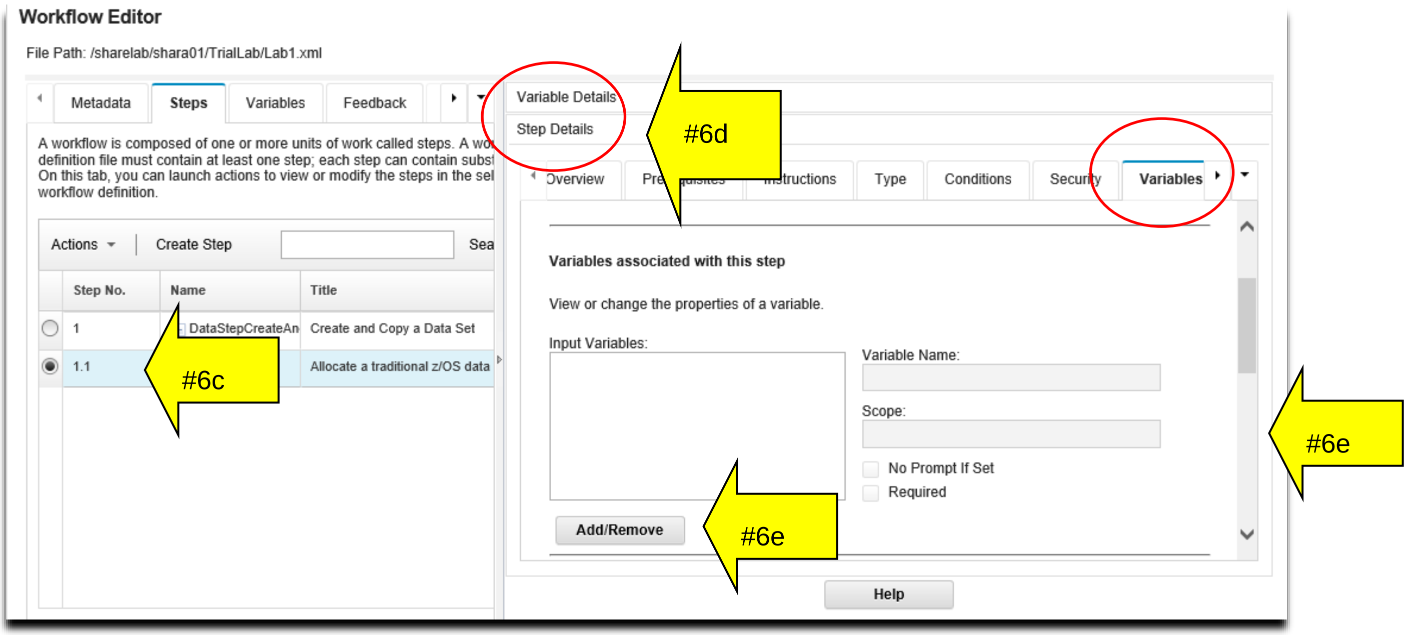
OK | Cancel

In your Workflow, you only have the one leaf step in a parent step you added left. Let's associate that leaf step with the two variables that we have defined.

c. Click on the leaf step **AllocDSN**, which is now step #1.1. (You might have to untwist the parent step to see it.)

d. Click on the right hand side's **Variable Details,** and then **Step Variables.**

e. Scroll down to the bottom, so you can see the **Add/Remove** button. Click on **Add/Remove.**



Workflow Editor

File Path: /sharelab/shara01/TrialLab/Lab1.xml

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

Step No.	Name	Title
1	DataStepCreateAn	Create and Copy a Data Set
1.1		Allocate a traditional z/OS data

Variables associated with this step

View or change the properties of a variable.

Input Variables:

Variable Name:

Scope:

No Prompt If Set

Required

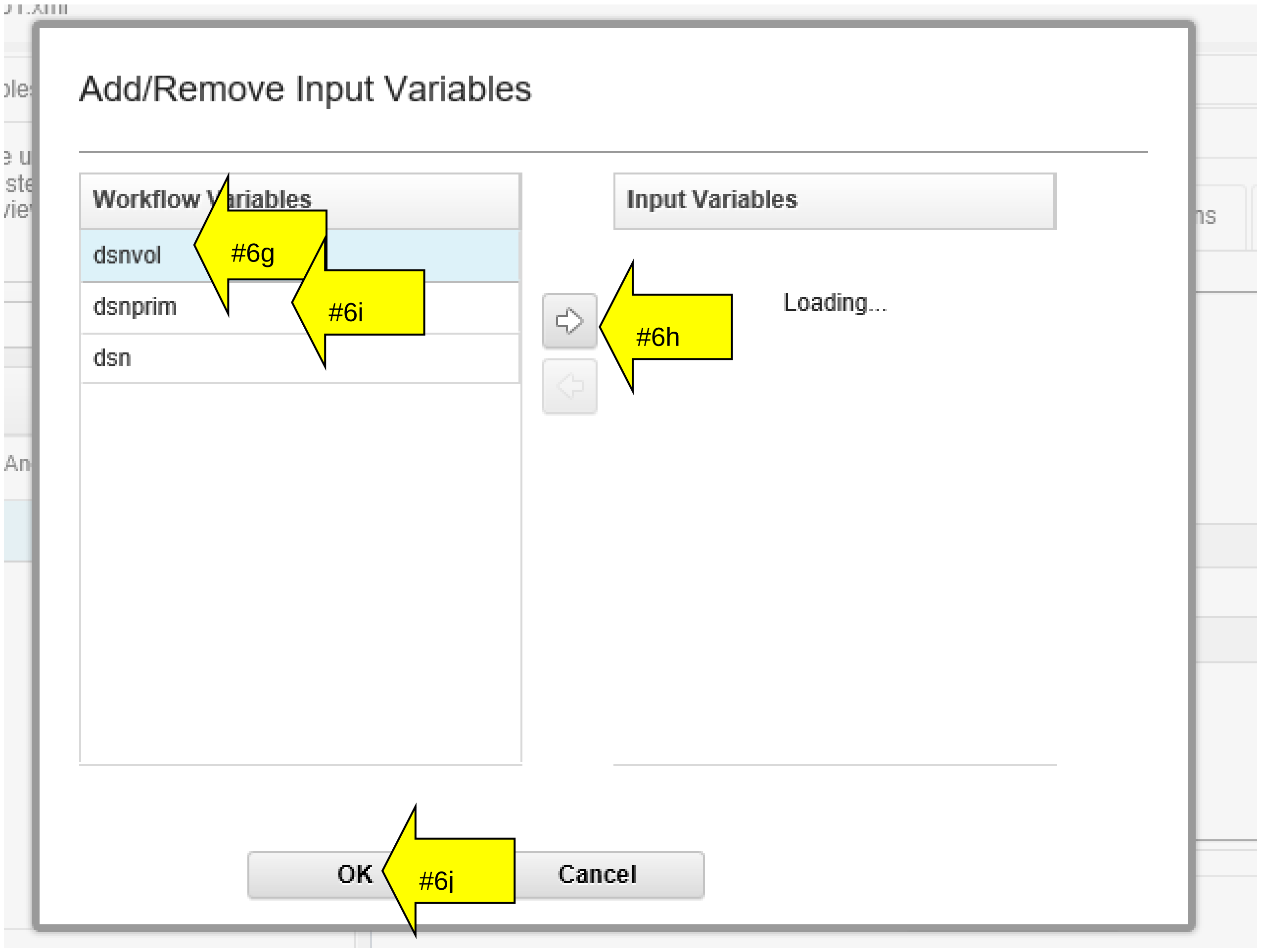
f. The **Add/Remove Input Variables** dialog appears. Look, there are the three variables that we added, but we need to move them from the **Workflow Variables** column into the **Input Variables** column.

g. Click on ***dsnvol*** first.

h. Then the **arrow** to move it to the other column.

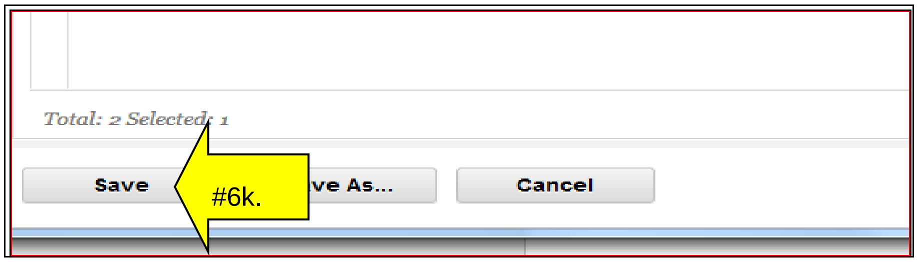
i. Repeat the same for the ***dsn*** and ***dsnprim*** variables.

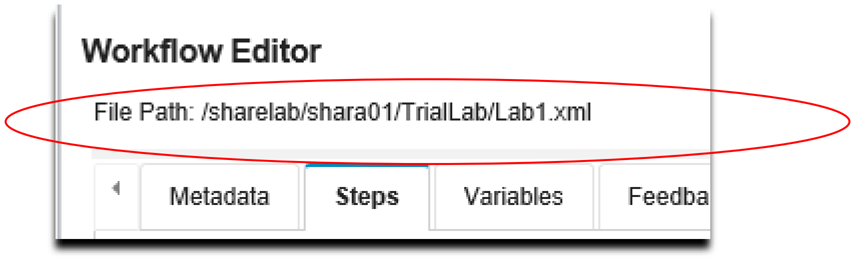
j. Click on **OK** to complete the dialog.



Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

k. 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 ***/sharelab/YourUserId/TrialLab/Lab1.xml*** remembering the get the case correct.***)***





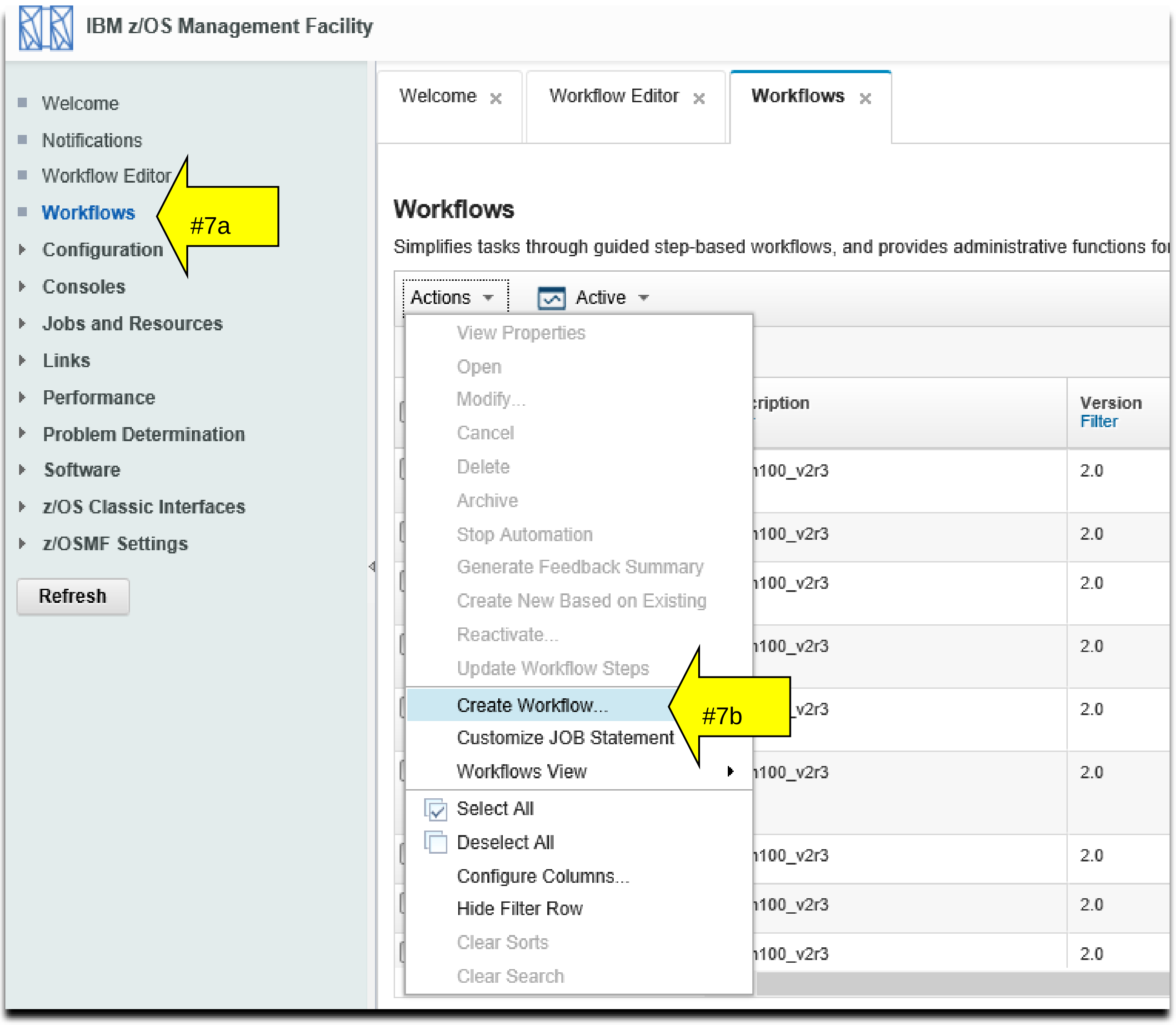
File Path: /sharelab/shara01/TrialLab/Lab1.xml

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!

a. Click on **Workflows** as the choice from the selections on the far left (below the **Workflow Editor).**

b. Under **Actions**, click on **Create Workflow.**

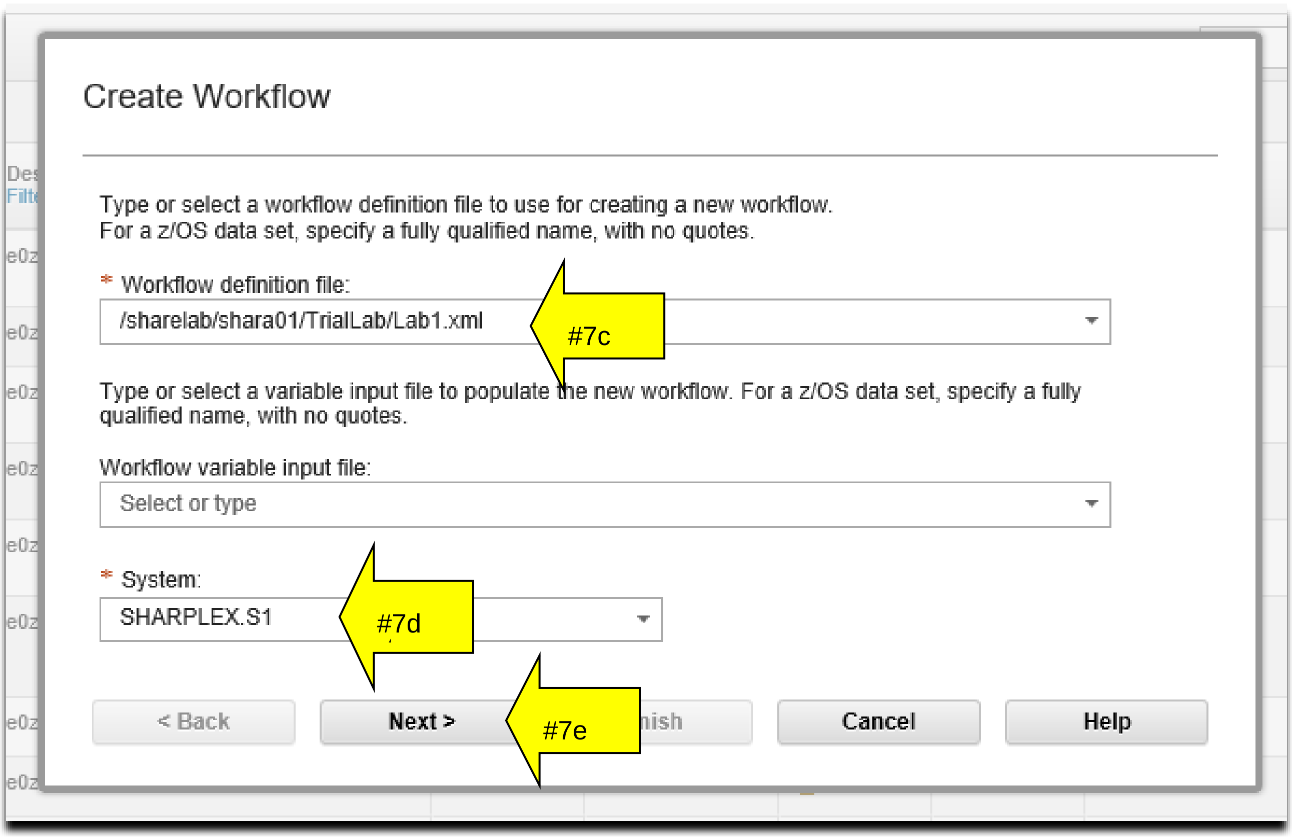


c. On the **Create Workflow** dialog, enter your Workflow definition file you just created. (which was ***/sharelab/YourUserId/TrialLab/Lab1.xml)*** where ***YourUserId*** was the userid you were given for the lab.

d. Select your **System** which is **SHARPLEX.S1** for this lab system.

e. Click on **Next.**

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.



Create Workflow

Type or select a workflow definition file to use for creating a new workflow.
For a z/OS data set, specify a fully qualified name, with no quotes.

* Workflow definition file:
/sharelab/shara01/TrialLab/Lab1.xml

Type or select a variable input file to populate the new workflow. For a z/OS data set, specify a fully qualified name, with no quotes.

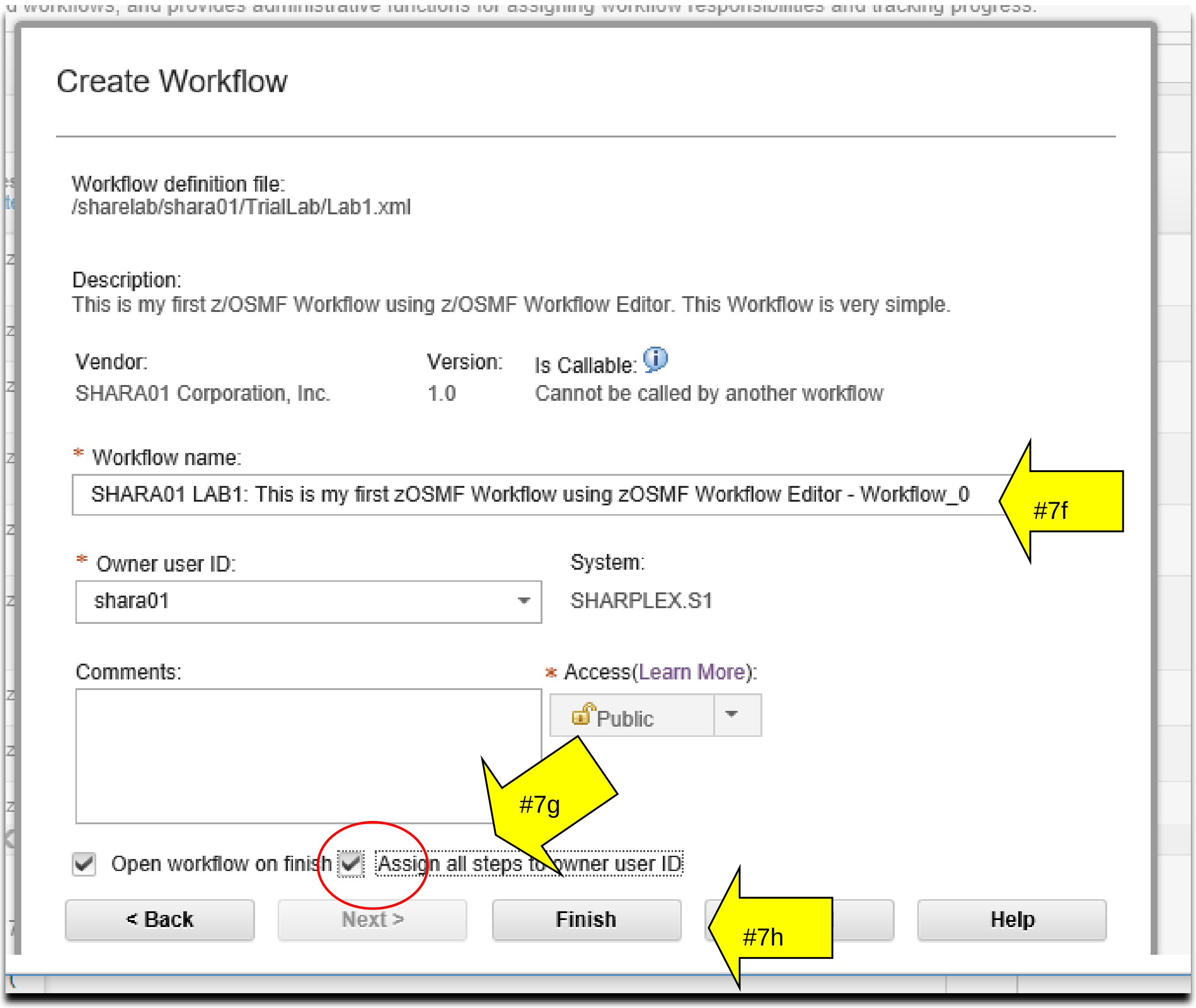
Workflow variable input file:
Select or type

* System:
SHARPLEX.S1

f. 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 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. -Workflow_0** 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).

g. 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.

h. Then click on **Finish.**



Create Workflow

Workflow definition file:
/sharelab/shara01/TrialLab/Lab1.xml

Description:
This is my first z/OSMF Workflow using z/OSMF Workflow Editor. This Workflow is very simple.

Vendor:	Version:	Is Callable:
SHARA01 Corporation, Inc.	1.0	Cannot be called by another workflow

* Workflow name: SHARA01 LAB1: This is my first zOSMF Workflow using zOSMF Workflow Editor - Workflow_0

* Owner user ID: shara01

System: SHARPLEX.S1

Comments:

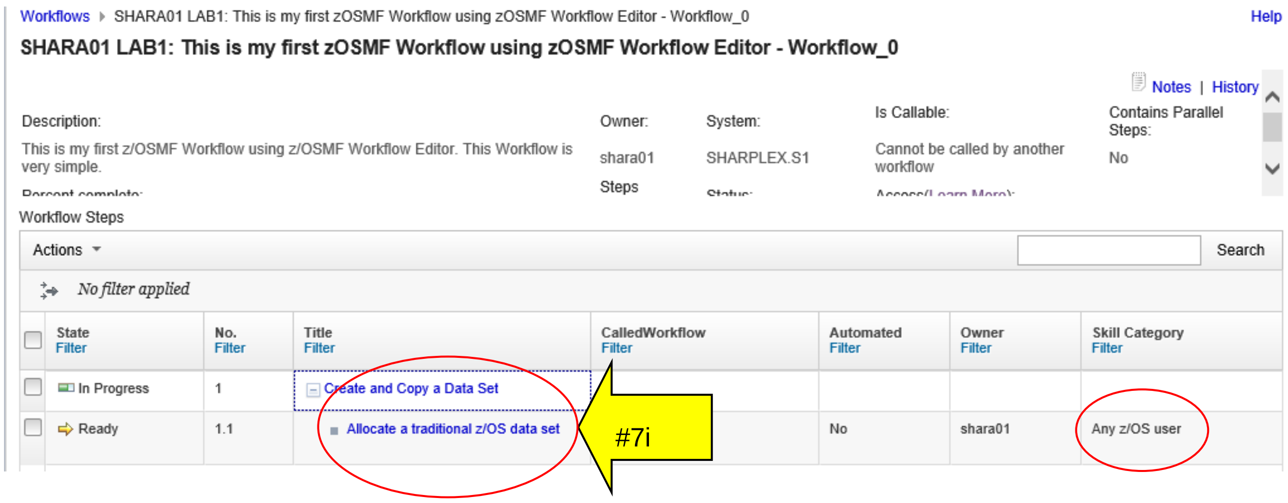
* Access(Learn More): Public

✓ Open workflow on finish ✓ Assign all steps to owner user ID

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Now, this should look familiar. You can see in this Workflow a parent step called “Create and Copy a Data Set”, one leaf step called “Allocate a traditional 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.**

i. Click on the blue **Allocate a traditional z/OS data set,** so we can try out our leaf step we created.



Workflows ▸ SHARA01 LAB1: This is my first zOSMF Workflow using zOSMF Workflow Editor - Workflow_0 Help

SHARA01 LAB1: This is my first zOSMF Workflow using zOSMF Workflow Editor - Workflow_0

Notes | History

Description: This is my first z/OSMF Workflow using z/OSMF Workflow Editor. This Workflow is very simple.

Owner: shara01

System: SHARPLEX.S1

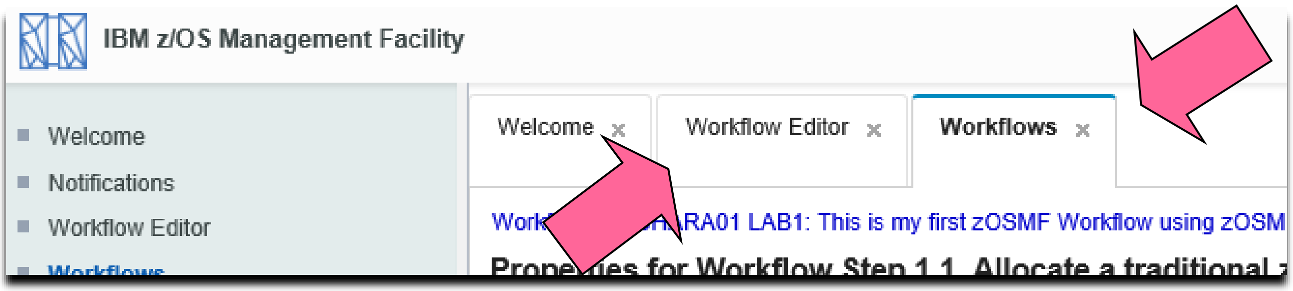
Is Callable: Cannot be called by another workflow

Contains Parallel Steps: No

Workflow Steps

State	No.	Title	CalledWorkflow	Automated	Owner	Skill Category
In Progress	1	Create and Copy a Data Set				
Ready	1.1	Allocate a traditional z/OS data set		No	shara01	Any z/OS user

Notice that at any time you can refer back to your Workflow definition, by just clicking on the z/OSMF tab for **Workflow Editor**, and then return to your Workflow by clicking back to the **Workflows** tab. 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.



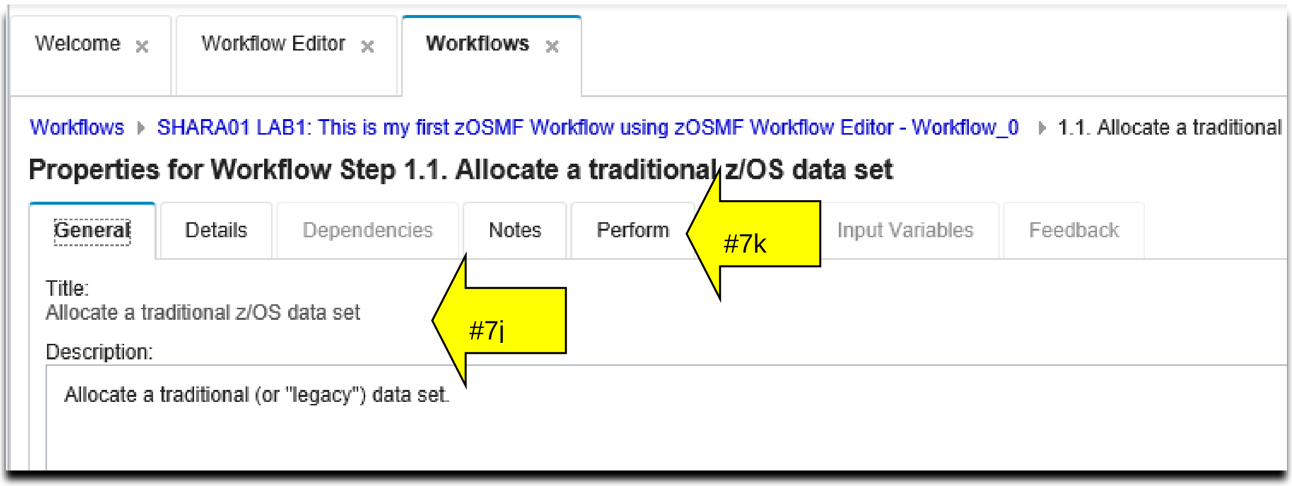
Work... SHARA01 LAB1: This is my first zOSMF Workflow using zOSM

Properties for Workflow Step 1.1 Allocate a traditional

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.



Workflows ▸ SHARA01 LAB1: This is my first zOSMF Workflow using zOSMF Workflow Editor - Workflow_0 ▸ 1.1. Allocate a traditional

Properties for Workflow Step 1.1. Allocate a traditional z/OS data set

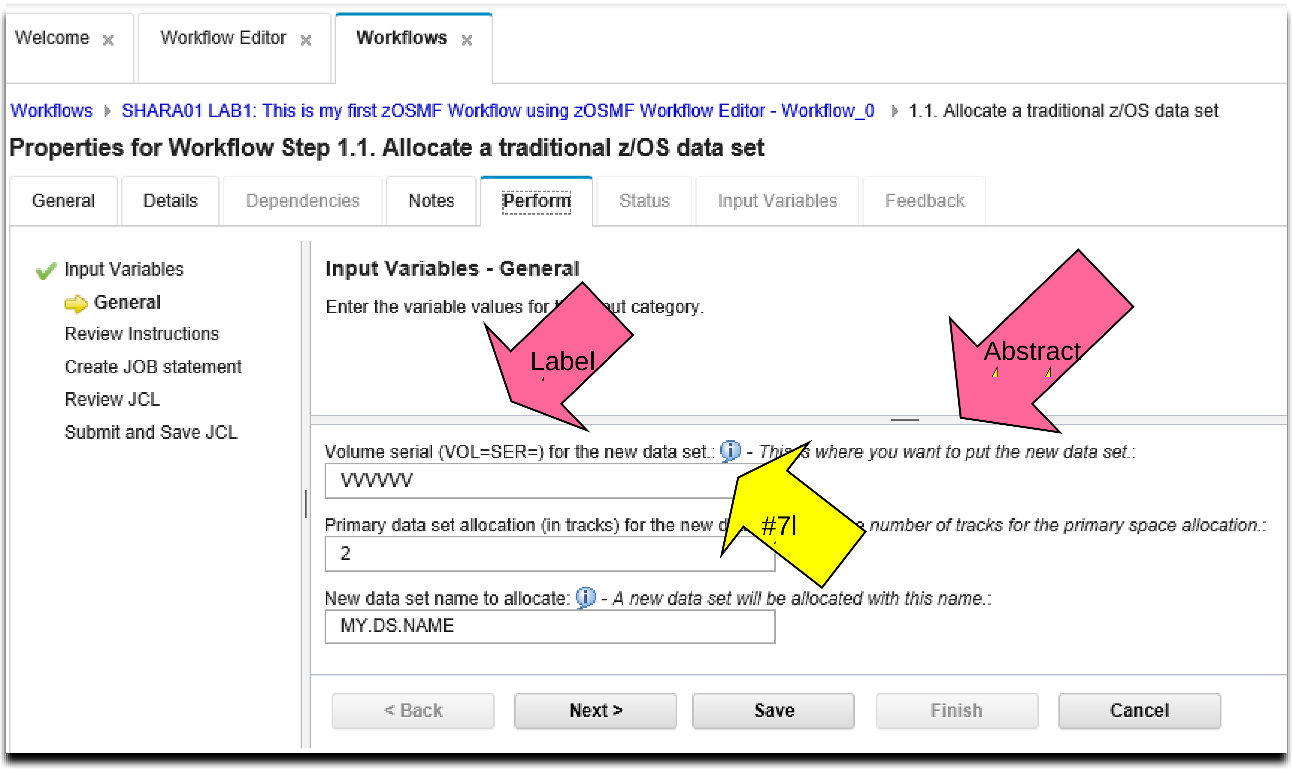
Title:
Allocate a traditional z/OS data set

Description:

Allocate a traditional (or "legacy") data set.

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

l. To Perform the step, the user is looking at the variables he'll need to supply to run this step. We see here the three variables we entered from the Editor, with the correct default values. The pink arrows are what Editor variable fields created those values. You don't get a choice on which variable appears first.



Properties for Workflow Step 1.1. Allocate a traditional z/OS data set

Input Variables - General

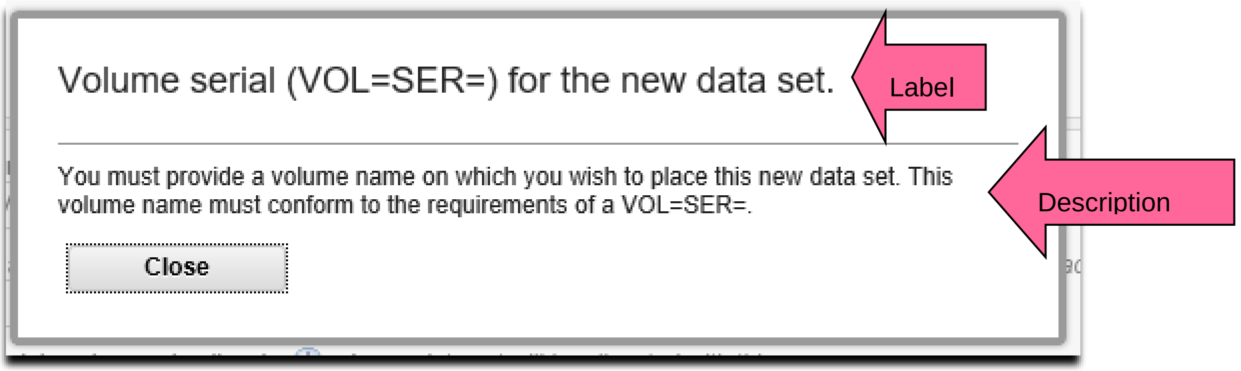
Enter the variable values for this input category.

Volume serial (VOL=SER=) for the new data set.: - This is where you want to put the new data set.:
VVVVVV

Primary data set allocation (in tracks) for the new data set.: - number of tracks for the primary space allocation.:
2

New data set name to allocate: - A new data set will be allocated with this name.:
MY.DS.NAME

Click on the little blue "i" in the bubble, and we can see this longer description. This is from the Editor: **Workflow Variables -> Label** (above the dividing line), and **Workflow Variables -> Description** (below the dividing line and above the **Close** button).



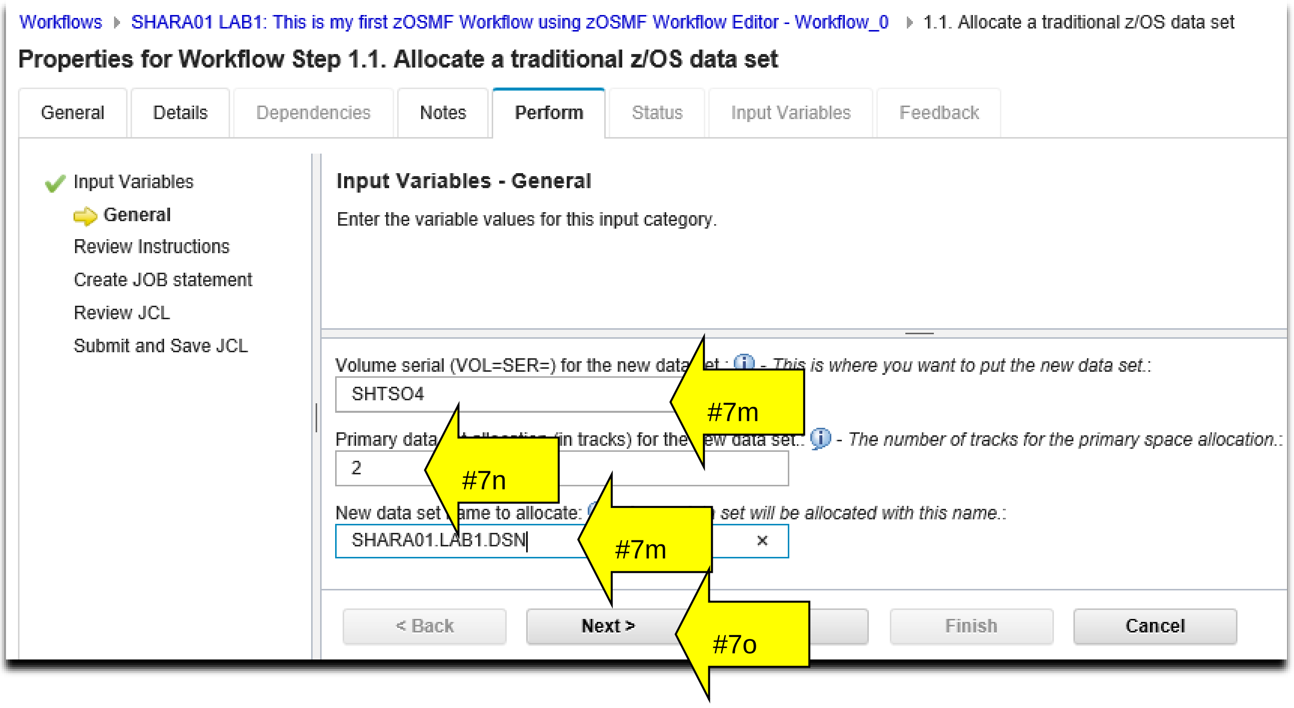
Volume serial (VOL=SER=) for the new data set.

You must provide a volume name on which you wish to place this new data set. This volume name must conform to the requirements of a VOL=SER=.

m. For this lab system, change the default volser and make it ***SHTSO4.*** (That is the letter O, not zero, all uppercase.) For this lab system, change the default data set name, and make it ***YourUserid*****.LAB1.DSN.**, all uppercase.

n. Leave the primary variable as the default (**2**). If you wanted to try to give an invalid number (such as 12) for the primary variable, you'll see that is it not accepted. The acceptable range for an integer variable would be a nice thing to put in an **Abstract** or maybe a **Description**, don't you think?

o. Then click on **Next>**


Workflows ▸ SHARA01 LAB1: This is my first zOSMF Workflow using zOSMF Workflow Editor - Workflow_0 ▸ 1.1. Allocate a traditional z/OS data set

Properties for Workflow Step 1.1. Allocate a traditional z/OS data set

Input Variables - General

Enter the variable values for this input category.

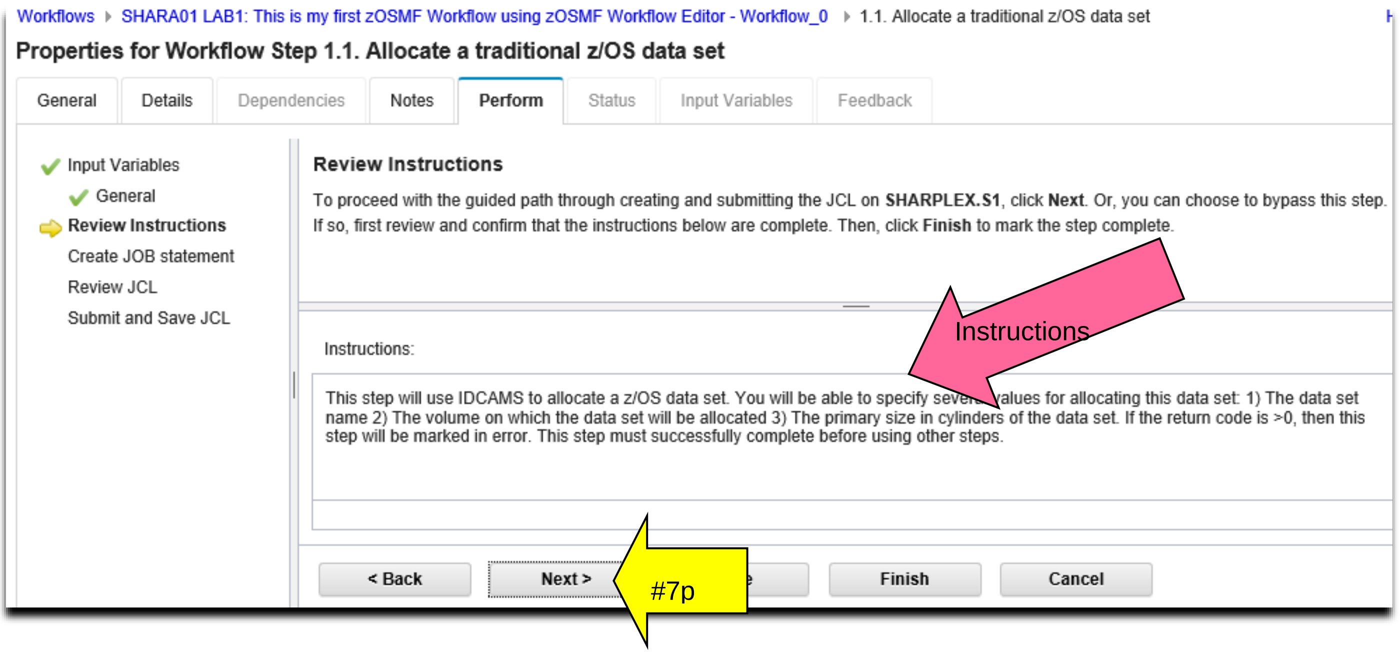
Volume serial (VOL=SER=) for the new data set: - This is where you want to put the new data set.: SHTSO4

Primary data set allocation (in tracks) for the new data set.: - The number of tracks for the primary space allocation.: 2

New data set name to allocate: - set will be allocated with this name.: SHARA01.LAB1.DSN

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

p. Recognize these instructions? These are what you put for **Step Details -> Instructions.** Click on **Next>.**



Properties for Workflow Step 1.1. Allocate a traditional z/OS data set

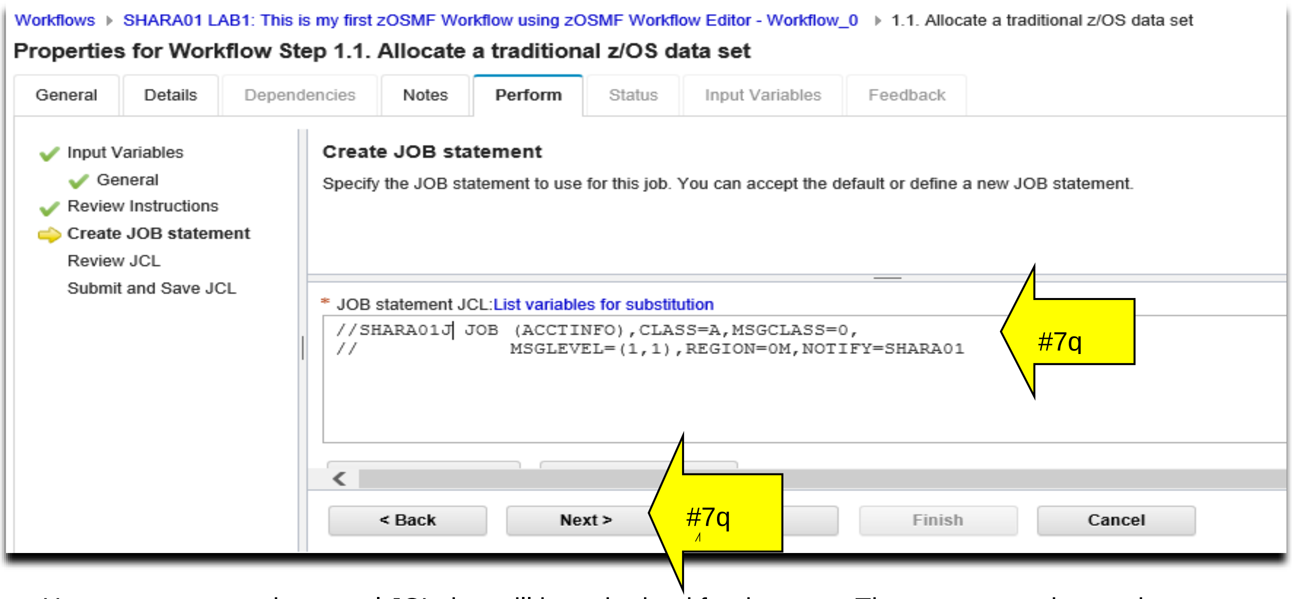
Review Instructions

To proceed with the guided path through creating and submitting the JCL on **SHARPLEX.S1**, click **Next**. Or, you can choose to bypass this step. If so, first review and confirm that the instructions below are complete. Then, click **Finish** to mark the step complete.

Instructions:

This step will use IDCAMS to allocate a z/OS data set. You will be able to specify several values for allocating this data set: 1) The data set name 2) The volume on which the data set will be allocated 3) The primary size in cylinders of the data set. If the return code is >0, then this step will be marked in error. This step must successfully complete before using other steps.

q. Now, you see the JOB information. You don't control the JOBCARD with anything in the Workflow Editor. The Workflow function handles this JOBCARD statement, and any updates you want to do with it. Change the JOB name to be what you want, and click on **Next>.**


Workflows ▸ SHARA01 LAB1: This is my first zOSMF Workflow using zOSMF Workflow Editor - Workflow_0 ▸ 1.1. Allocate a traditional z/OS data set

Properties for Workflow Step 1.1. Allocate a traditional z/OS data set

Create JOB statement

Specify the JOB statement to use for this job. You can accept the default or define a new JOB statement.

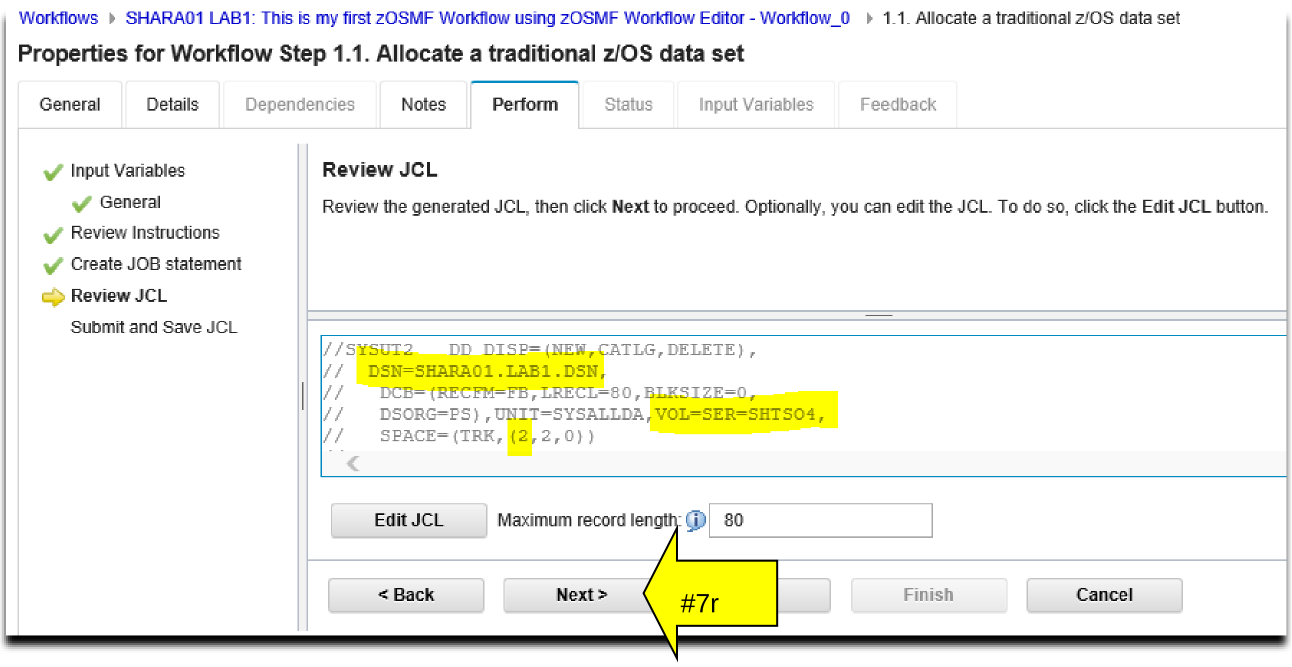
* JOB statement JCL: List variables for substitution

```
//SHARA01J JOB  (ACCTINFO),CLASS=A,MSGCLASS=0,
//             MSGLEVEL=(1,1),REGION=0M,NOTIFY=SHARA01
```

r. Here, you can see the actual JCL that will be submitted for the step. There are some interesting things to note:

- **DSN=** and **VOL=SER=** correctly reflect the default override the user gave us!
- And that the default for **`SPACE=(TRK,(2`** was picked up correctly.

Then click on **Next >.**


Workflows ▸ SHARA01 LAB1: This is my first zOSMF Workflow using zOSMF Workflow Editor - Workflow_0 ▸ 1.1. Allocate a traditional z/OS data set

Properties for Workflow Step 1.1. Allocate a traditional z/OS data set

Review JCL

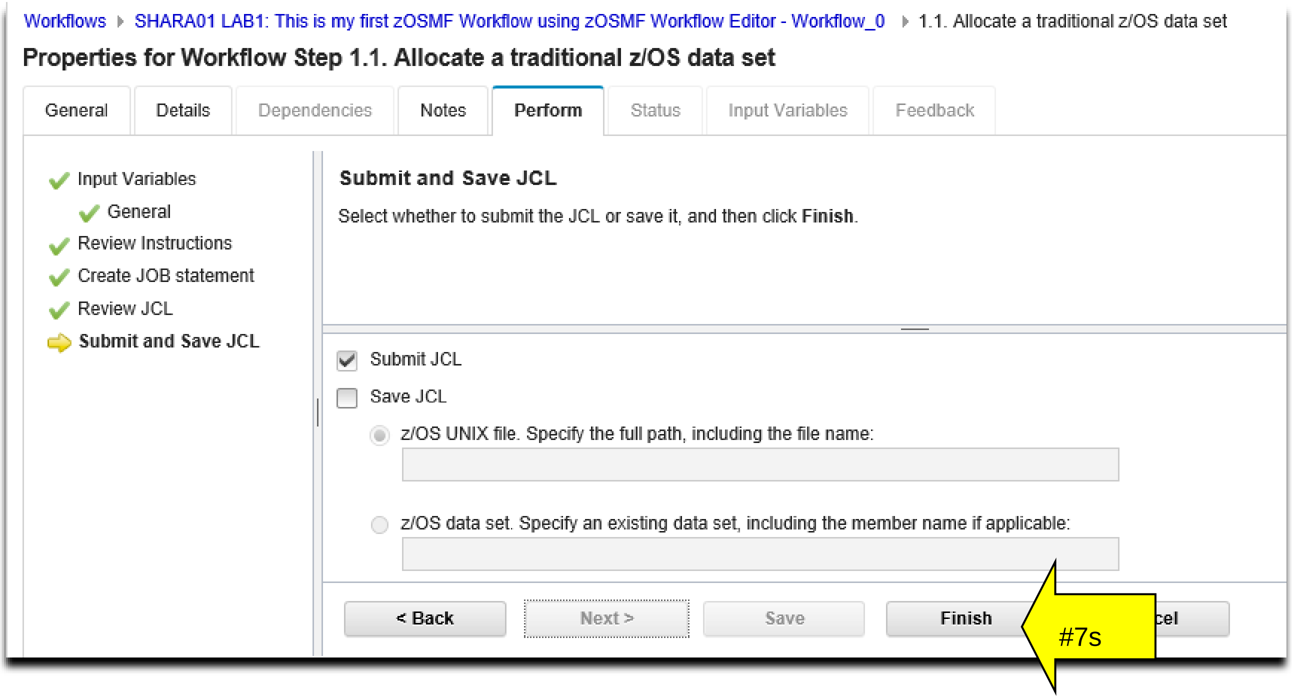
Review the generated JCL, then click **Next** to proceed. Optionally, you can edit the JCL. To do so, click the **Edit JCL** button.

```
//SYSUT2    DD DISP=(NEW,CATLG,DELETE),
//  DSN=SHARA01.LAB1.DSN,
//    DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,
//    DSORG=PS),UNIT=SYSALLDA,VOL=SER=SHTS04,
//    SPACE=(TRK,(2,2,0))
```

Maximum record length: 80

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

s. Moving through, let’s submit this JCL and move onto some more advanced topics if you have time. Click on **Finish.** You can click on the **Refresh** button to see the output from the job, if you don’t see the output tabs right away.



Workflows ▸ SHARA01 LAB1: This is my first zOSMF Workflow using zOSMF Workflow Editor - Workflow_0 ▸ 1.1. Allocate a traditional z/OS data set

Properties for Workflow Step 1.1. Allocate a traditional z/OS data set

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

- ✓ Input Variables
  - ✓ General
- ✓ Review Instructions
- ✓ Create JOB statement
- ✓ Review JCL
- ⇨ **Submit and Save JCL**

Submit and Save JCL

Select whether to submit the JCL or save it, and then click **Finish**.

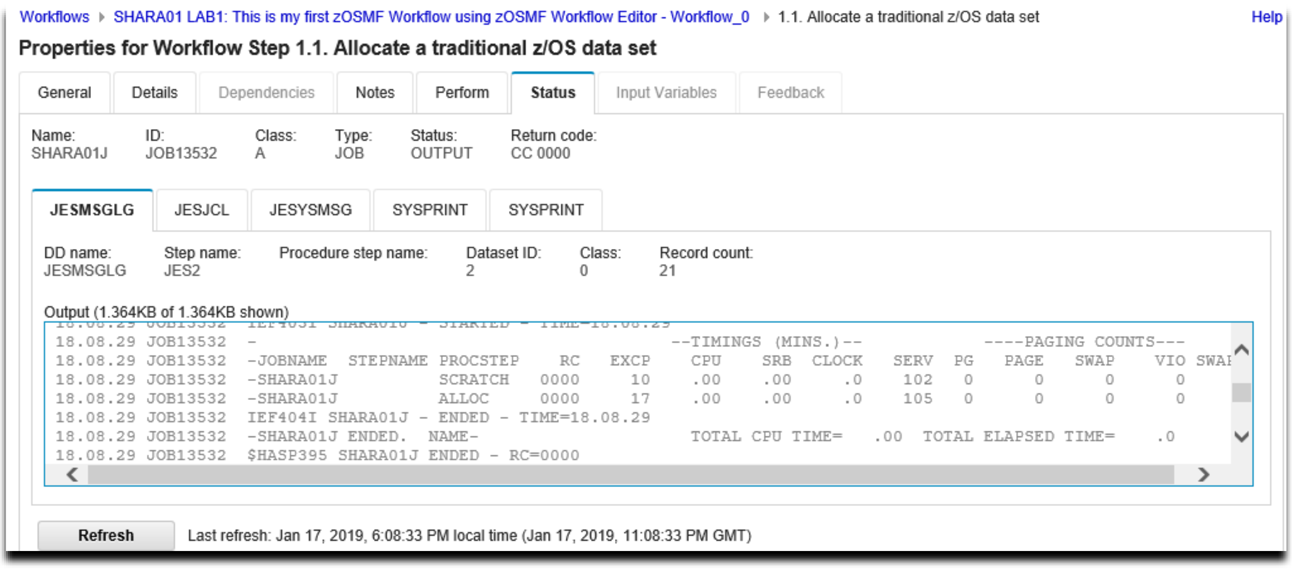
[✓] Submit JCL

[ ] Save JCL

○ z/OS UNIX file. Specify the full path, including the file name:

○ z/OS data set. Specify an existing data set, including the member name if applicable:

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



Workflows ▸ SHARA01 LAB1: This is my first zOSMF Workflow using zOSMF Workflow Editor - Workflow_0 ▸ 1.1. Allocate a traditional z/OS data set Help

Properties for Workflow Step 1.1. Allocate a traditional z/OS data set

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

Name:	ID:	Class:	Type:	Status:	Return code:
SHARA01J	JOB13532	A	JOB	OUTPUT	CC 0000

JESMSGLG | JESJCL | JESYSMSG | SYSPRINT | SYSPRINT

DD name:	Step name:	Procedure step name:	Dataset ID:	Class:	Record count:
JESMSGLG	JES2		2	0	21

Output (1.364KB of 1.364KB shown)

```
18.08.29 JOB13532  -                                         --TIMINGS (MINS.)--            ----PAGING COUNTS---
18.08.29 JOB13532  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV  PG   PAGE   SWAP    VIO SWAI
18.08.29 JOB13532  -SHARA01J          SCRATCH   0000     10    .00    .00     .0    102   0      0      0      0
18.08.29 JOB13532  -SHARA01J          ALLOC     0000     17    .00    .00     .0    105   0      0      0      0
18.08.29 JOB13532  IEF404I SHARA01J - ENDED - TIME=18.08.29
18.08.29 JOB13532  -SHARA01J ENDED.  NAME-                     TOTAL CPU TIME=   .00  TOTAL ELAPSED TIME=    .0
18.08.29 JOB13532  $HASP395 SHARA01J ENDED - RC=0000
```

Last refresh: Jan 17, 2019, 6:08:33 PM local time (Jan 17, 2019, 11:08:33 PM GMT)

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

1. Create a brand new Workflow
2. Add variables to the Workflow, so that the user can provide unique information. How to specify certain types of variables for the Workflow to validate: data set names, volsers, and integers.
3. Add a parent step and a leaf step.
4. Associate the variables with the Workflow steps.
5. Run JCL in batch from a Workflow. (And how to code that JCL for the Velocity engine variables)
6. Save your Workflow definition file.
7. 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).

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

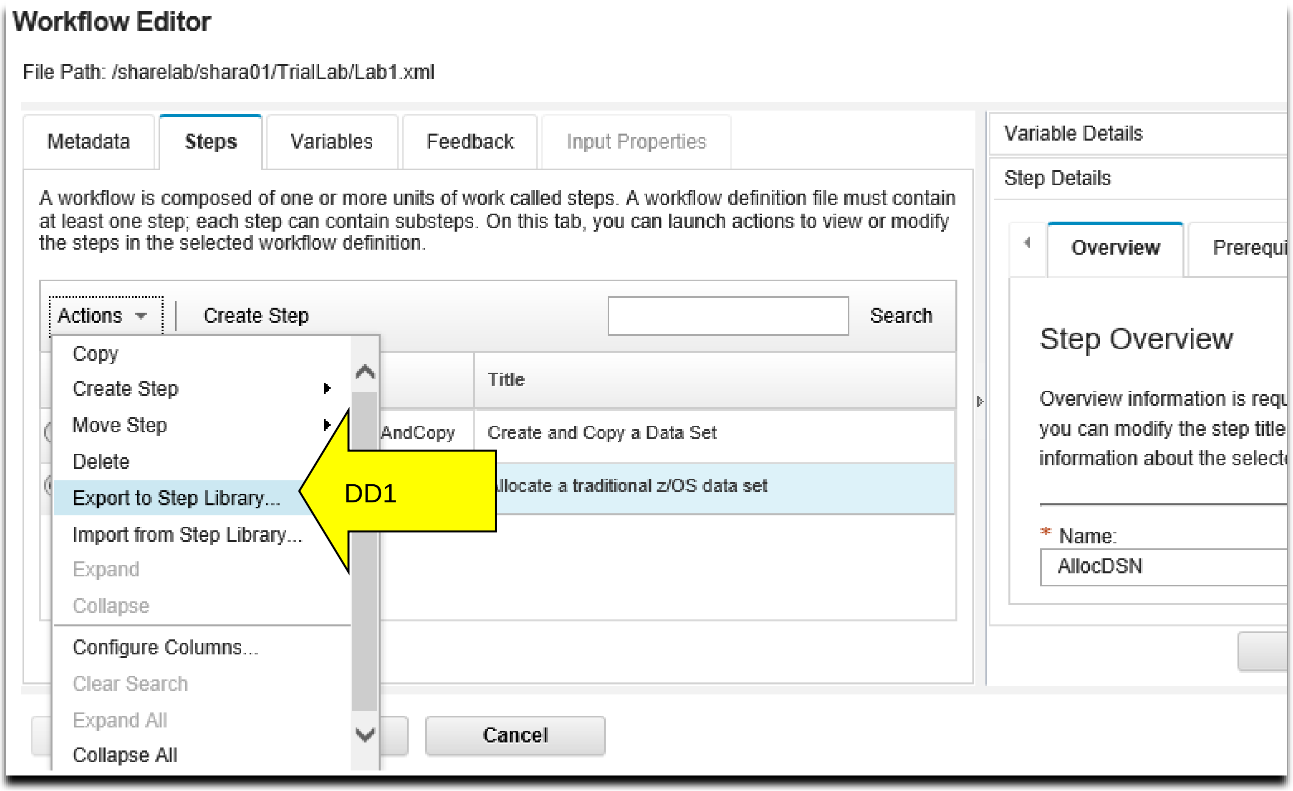
Deeper Dive

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 ***/sharelab/YourUserid/*TrialLab/Lab1.xml.**

Click on **Steps**, and click on step 1.1 (AllocDSN) 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…**


Workflow Editor

File Path: /sharelab/shara01/TrialLab/Lab1.xml

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.

	Title
AndCopy	Create and Copy a Data Set
	llocate a traditional z/OS data set

Step Overview

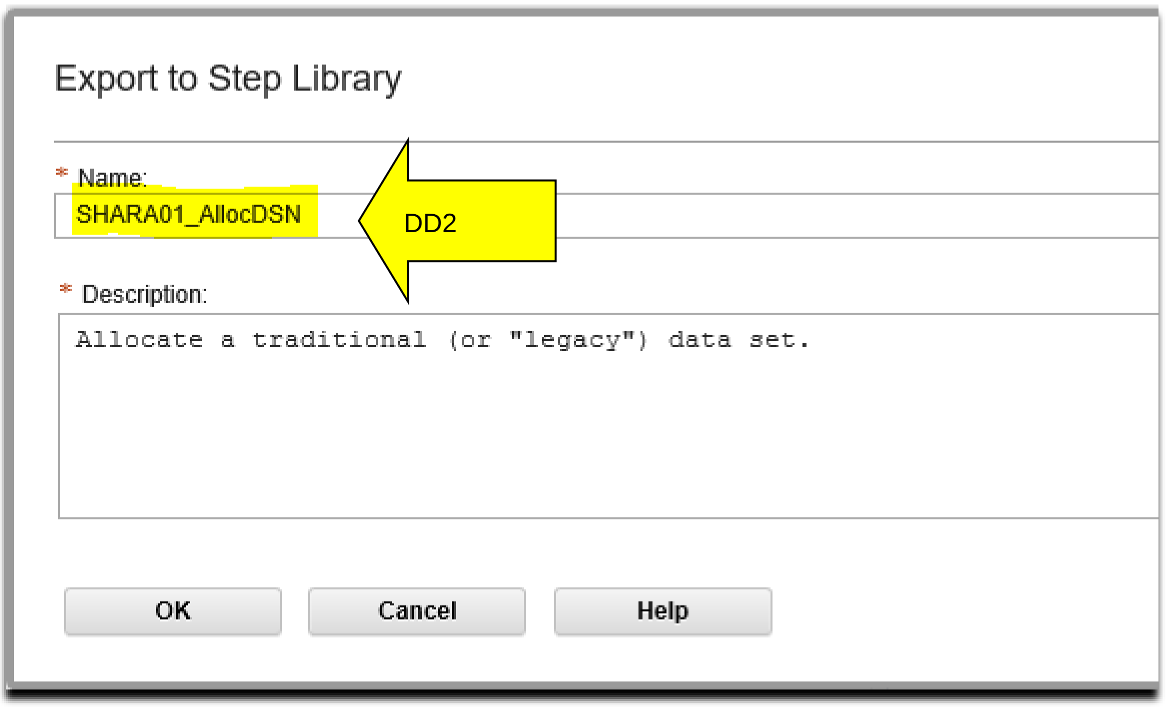
Overview information is requ you can modify the step title information about the select

* Name: AllocDSN

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*_AllocDSN.** 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.**


Export to Step Library

* Name:

SHARA01_AllocDSN

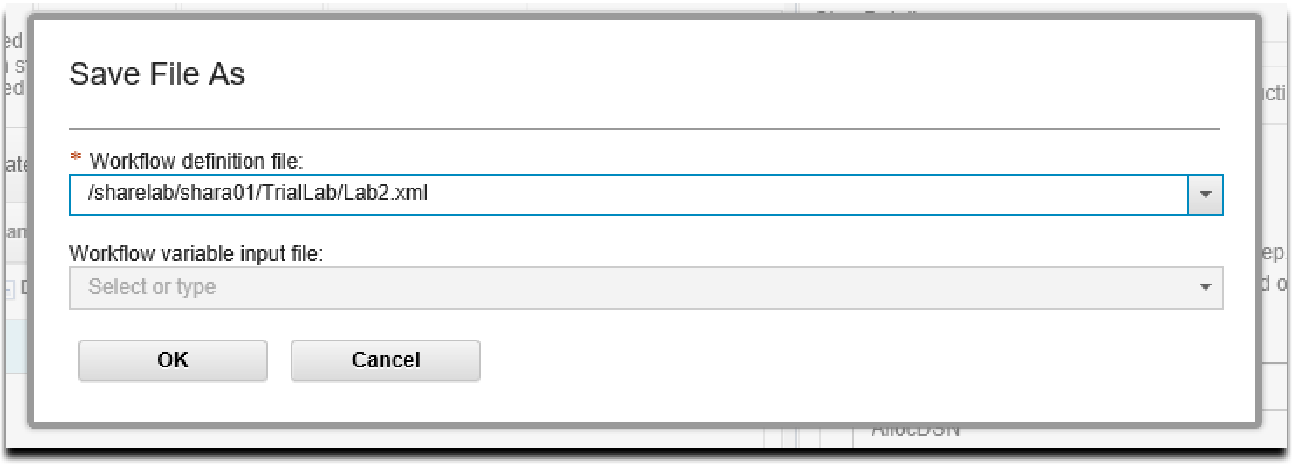
* Description:

```
Allocate a traditional (or "legacy") data set.
```

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

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

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 **/sharelab/*YourUserID*/TrialLab/Lab2.xml**. Click on **OK.**

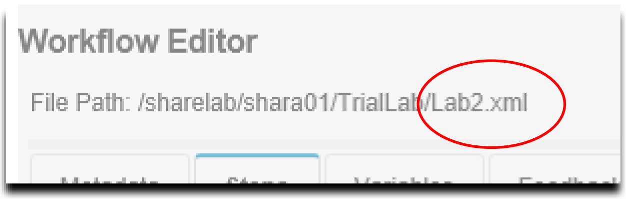


Save File As

* Workflow definition file: /sharelab/shara01/TrialLab/Lab2.xml

Workflow variable input file: Select or type

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

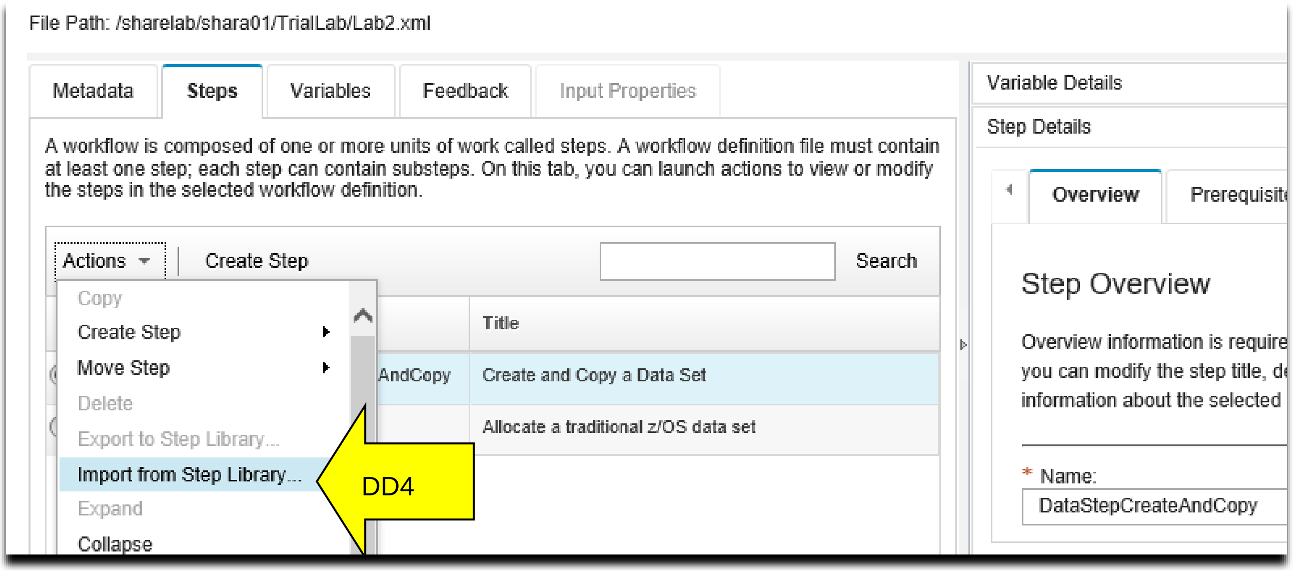


Workflow Editor

File Path: /sharelab/shara01/TrialLab/Lab2.xml

DD4: Let's add an exported "new" step to the "Lab2" Workflow definition file you are currently in. This how you can add an exported step into an existing Workflow. Since your changes are going into Lab2.xml, you don't have to worry if you ruin it, as Lab1.xml was already saved in a different Workflow definition file.

- To create a new step, **make sure step 1 (the parent step) is selected, and is highlighted.**
- We are going to pull from our Step Library. Go to **Actions** pull down, and select **Import from Step Library…**



File Path: /sharelab/shara01/TrialLab/Lab2.xml

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.

Title
Create and Copy a Data Set
Allocate a traditional z/OS data set

Step Overview

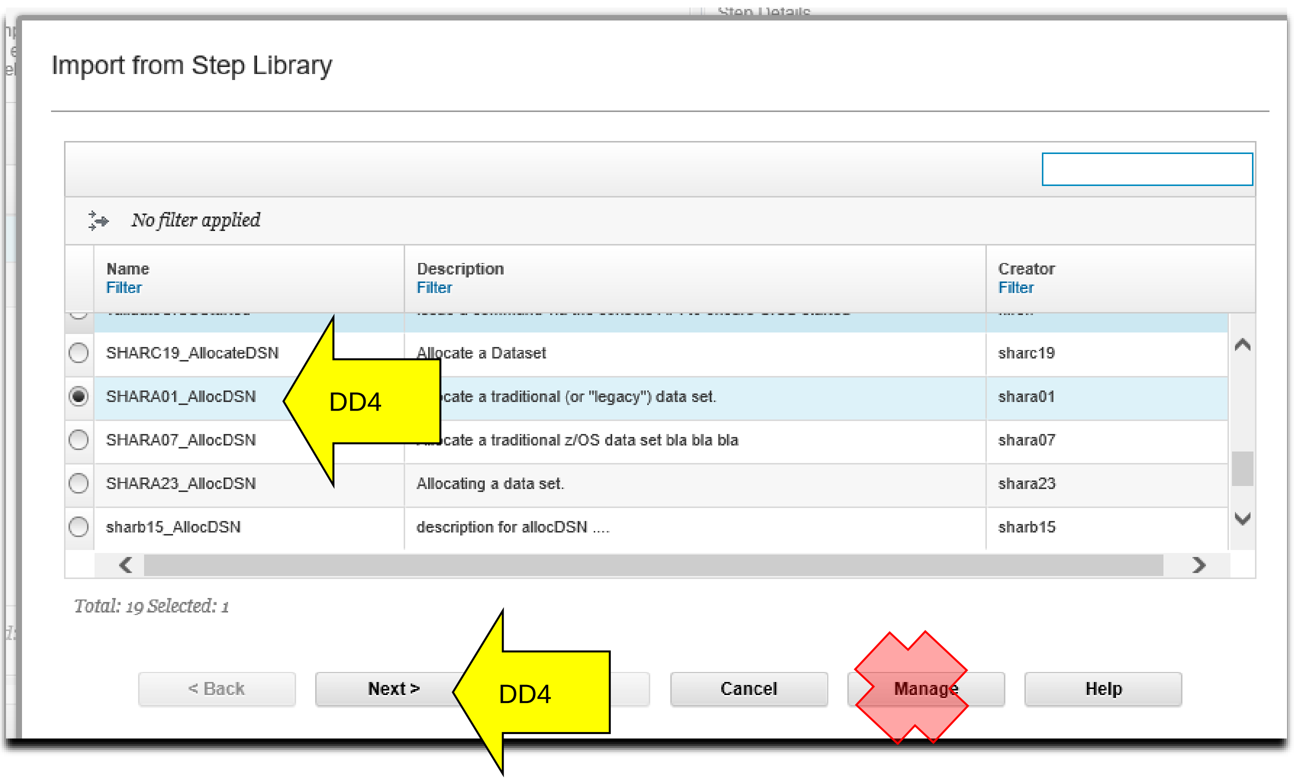
Overview information is require
you can modify the step title, de
information about the selected

* Name: DataStepCreateAndCopy

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-*** **_AllocDSN**), 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.

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.



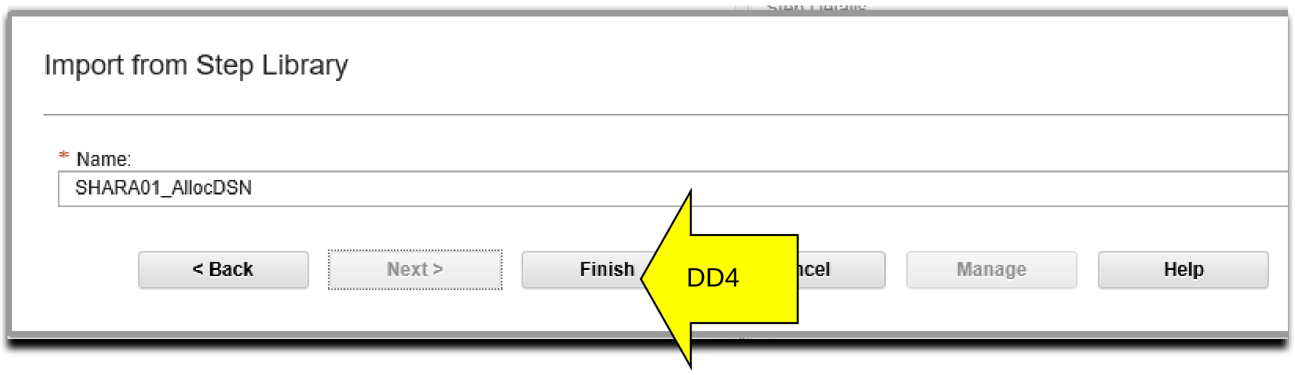
Import from Step Library

No filter applied

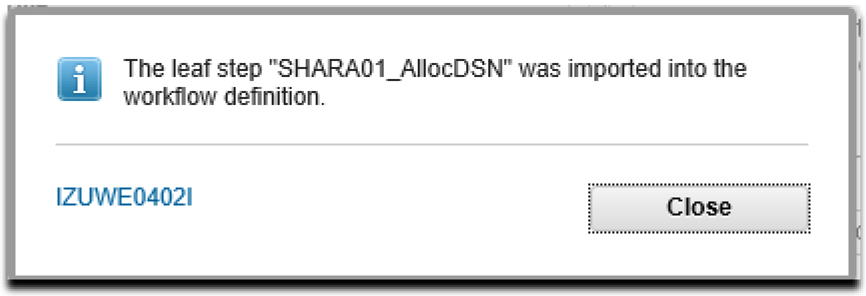
	Name	Description	Creator
○	SHARC19_AllocateDSN	Allocate a Dataset	sharc19
●	SHARA01_AllocDSN	Allocate a traditional (or "legacy") data set.	shara01
○	SHARA07_AllocDSN	Allocate a traditional z/OS data set bla bla bla	shara07
○	SHARA23_AllocDSN	Allocating a data set.	shara23
○	sharb15_AllocDSN	description for allocDSN	sharb15

Total: 19 Selected: 1

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



Success! Click on **Close.**


The leaf step "SHARA01_AllocDSN" was imported into the workflow definition.

IZUWE0402I

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**.

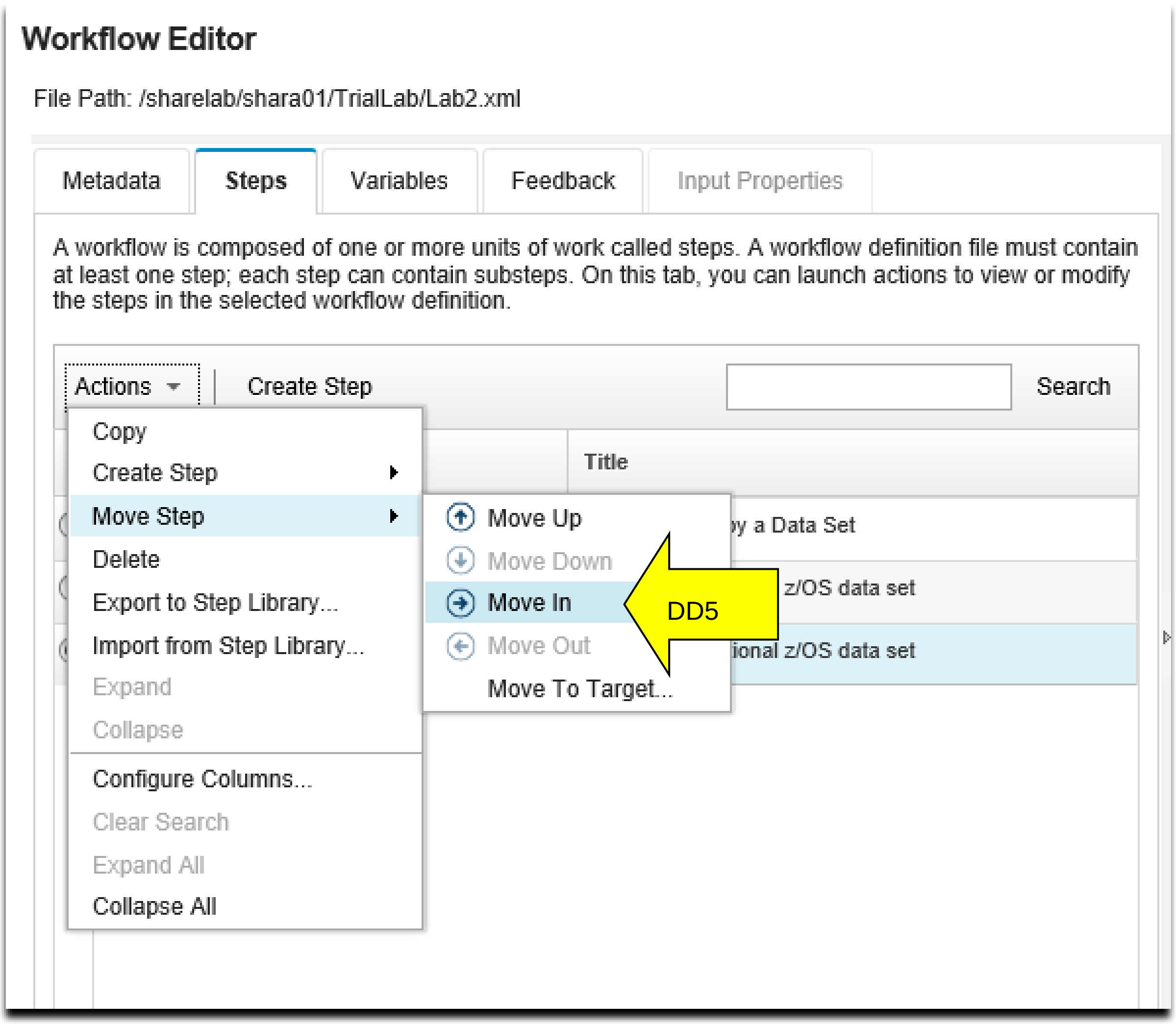
During unmarshalling, there is a logical "break" in connection between the workflow definition and the external referenced XML files. The unmarshalling process creates a single workflow definition file as output.

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.

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Moving steps around

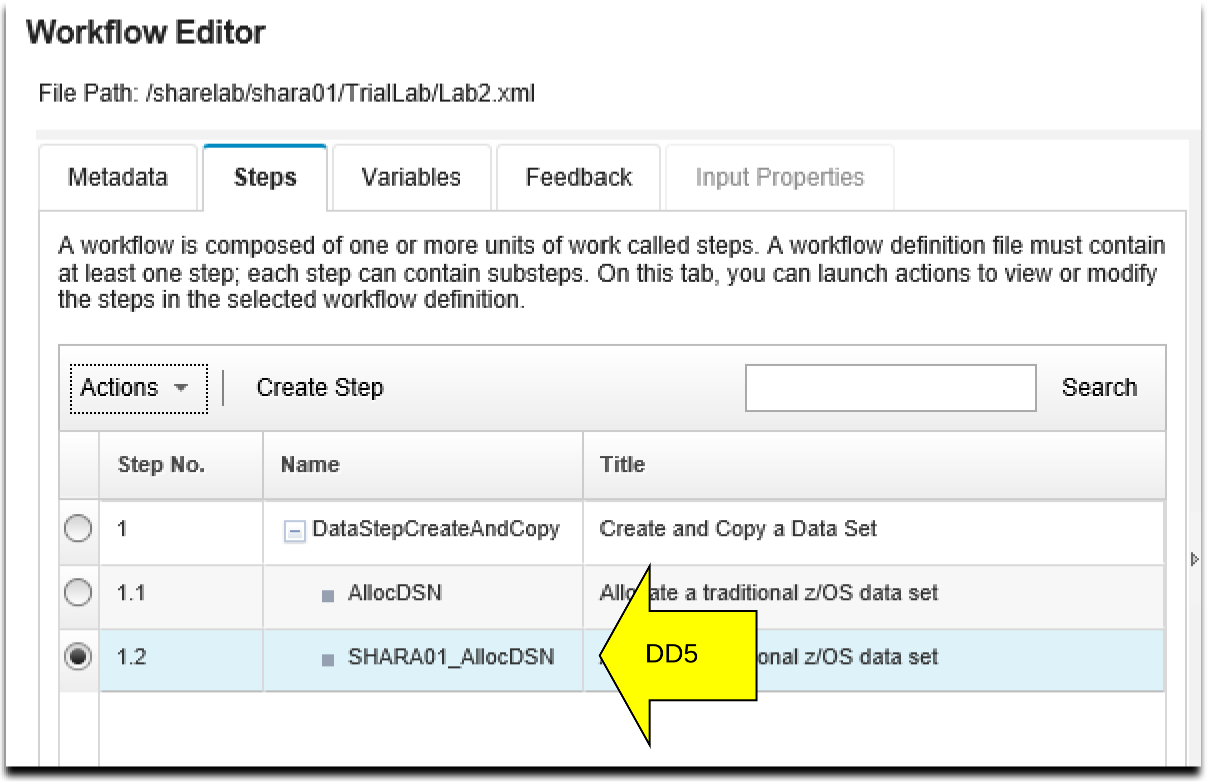
DD5: At this point, you have the step added after the parent step you selected. For this deep dive, let's pretend that we wanted this imported step to be prior parent step in our Workflow. It's easy to do, just click on the new step, **Actions,** then **Move Step >,** then **Move In.**


Workflow Editor

File Path: /sharelab/shara01/TrialLab/Lab2.xml

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.

You can now see it's moved up into the parent step, and is now Step 1.2 in the Workflow.



Workflow Editor

File Path: /sharelab/shara01/TrialLab/Lab2.xml

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	DataStepCreateAndCopy	Create and Copy a Data Set
○	1.1	AllocDSN	Allo ate a traditional z/OS data set
●	1.2	SHARA01_AllocDSN	onal z/OS data set

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

Putting it all together

If you still have time on your own, save your current Workflow as
/sharelab/*YourUserID*/TrailLab/Lab3.xml. Remember the Wofklow Editor will not let you create an invalid Workflow, so you might want to play around and see what you can try in this separate file.

Here are some suggestions for what to do:

1. Currently, you have two steps that allocate a data set. Remove the ***YourUserID*** **_Allocdsn** step (Step 1.2) You don't need this step as we used it as an example, but you don't need two allocation steps.
2. You can manually create (not import) a new step called **copydsn.** You will use the template that is found in **/shareuser/mwalle/EditorFiles/copydsn .** It is JCL, and will use DFSMSdss to copy from one data set to another, and uses variables for the "from" and "to" data set names, and a "to" volume.
   a. Put it as a new leaf step, under the parent step "**Create and Copy a Data Set**" It will be the last of the leaf steps under this parent step, and will be the new Step 1.2.
   b. We haven't shown this, but it is intuitive now…make the step **allocdsn** a precursor step to **copydsn.** This means that you have to have successful done **allocdsn,** to perform **copydsn.** (Hint: **Step Details -> Prerequisite)**
   c. You should now have one parent step, with two leaf steps under it: first to allocate, and then one to copy.
3. You will need to create one new variable which this step needs. Can you see in the template contents what the variables are called? (Hint: look for the Velocity engine $instance that is a new variable you haven't seen thus far.) You've already got two of the variables defined, and you'll need to add one more. Make sure you've got all the variables associated with this new step ☺.
4. You can import a step that is in the library called **listcat.** This step just does an IDCAMS LISTCAT for a variable called **dsn.** Something to think about: this is the same variable name that we already have definied in this Workflow (for steps 1.1 and 1.2). Does this matter?

Be careful! When you try to test your new Lab3.xml file, you might see this error:

```
A new instance of the workflow cannot be created because an instance already
exists. Only one instance of this workflow can be active at a time with the
workflow ID "YourUserid Workflow Editor Lab" and scope "system" .
```

In order to avoid this, you can either:

- change the **Metadata -> Workflow ID** for Lab3.xml to be a different name than what you had used for Lab1.xml or Lab2.xml. It's not obvious, but that is what this is saying (preferred).
- -or- delete your Workflows you created from Lab1.xml or Lab2.xml

Appendix of samples provided for this lab

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

```
EDIT       /shareuser/mwalle/EditorFiles/allocdsn          Columns 00001 00072
Command ===>                                                  Scroll ===> HALF
****** ***************************** Top of Data *****************************
000001 //SCRATCH  EXEC PGM=IDCAMS
000002 //SYSPRINT DD SYSOUT=*
000003 //SYSIN    DD *
000004    DELETE -
000005      $instance-dsn
000006    SET MAXCC=0
000007 //*
000008 //ALLOC EXEC PGM=IEBGENER
000009 //SYSPRINT DD   SYSOUT=*
000010 //SYSUT1   DD *
000011   z/OSMF Workflows can do lots of stuff!
000012 //SYSUT2   DD DISP=(NEW,CATLG,DELETE),
000013 //   DSN=$instance-dsn,
000014 //    DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,
000015 //    DSORG=PS),UNIT=SYSALLDA,VOL=SER=$instance-dsnvol,
000016 //    SPACE=(TRK,($instance-dsnprim,2,0))
000017 //SYSIN    DD   DUMMY
000018 //*
****** **************************** Bottom of Data ****************************
```

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.

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

```
EDIT       /shareuser/mwalle/EditorFiles/copydsn           Columns 00001 00072
 Command ===>                                                  Scroll ===> HALF
 ****** ***************************** Top of Data ****************************
 000001 //DSS     EXEC  PGM=ADRDSSU,REGION=0M
 000002 //SYSPRINT  DD SYSOUT=*
 000003 //TGTOUT1  DD VOL=SER=$instance-dsnvol,DISP=SHR,UNIT=SYSALLDA
 000004 //SYSIN    DD *
 000005  COPY DATASET(INCLUDE($instance-dsn)) -
 000006    OUTDD(TGTOUT1) -
 000007    TGTALLOC(SOURCE) -
 000008    REPLACEUNCONDITIONAL -
 000009    RENAMEU(($instance-dsn,$instance-outdsn)) -
 000010    TOLERATE(ENQFAILURE)
 ****** **************************** Bottom of Data **************************
```

Here is the sample of the **listcat** JCL, which is an exported step. Notice all the **$instance-*variable*** locations for the Velocity engine.

```
EDIT       /shareuser/mwalle/EditorFiles/listcat           Columns 00001 00072
Command ===>                                                  Scroll ===> HALF
****** ***************************** Top of Data ****************************
000001 //DSS     EXEC  PGM=IDCAMS,REGION=0M
000002 //SYSPRINT  DD SYSOUT=*
000003 //SYSIN    DD *
000004   LISTCAT ALL ENT($instance-dsn)
****** **************************** Bottom of Data **************************
```

Materials may not be reproduced in whole or in part
without the prior written permission of IBM.