DSM1

Not Just Members Only

How Data Gets Down on z/OS

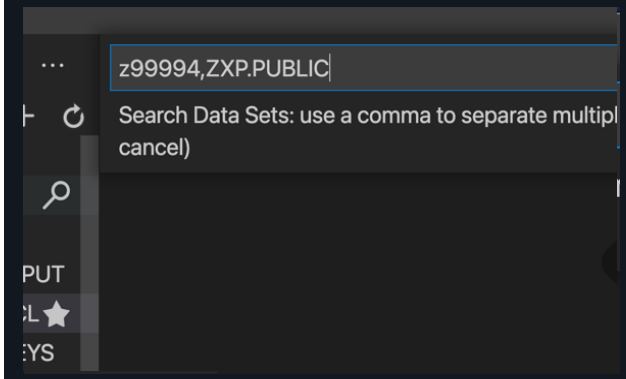
12 steps 45 minutes

THE CHALLENGE

As you've seen, data gets handled a little differently on z/OS. This isn't just Z trying to be difficult. When records are organized and kept in a structure that lines up with how they will typically be read and written, applications can run faster, and with less confusion. In this challenge, we'll get our hands on some data sets and members, and then learn how to copy, rename and even delete them like it's no big deal.

BEFORE YOU BEGIN

If you have access to the system, you're all good to go. The only technical challenged you need to have completed before this one is VSC1.

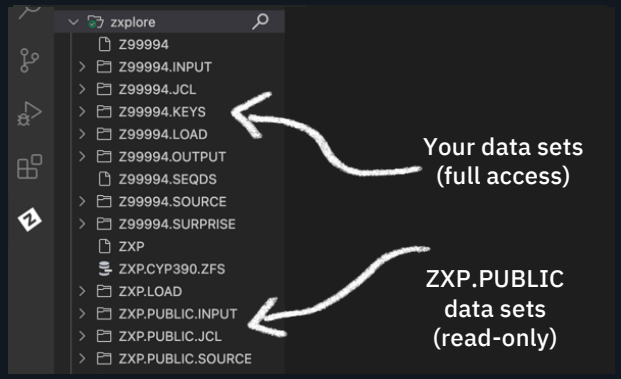


1. REFINE YOUR FILTER

We're going to expand our filter to see even more data sets.

Click on the Magnifying Glass to the right of your profile name, and enter the following:
ZXXXXX,ZXP.PUBLIC
(Please make sure to enter your own userid here, not Z99994)

Also, make sure to note the comma before ZXP and the period after it. These are important.

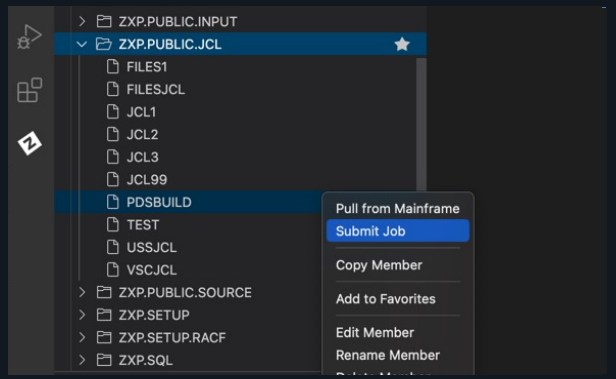


2. YOU CAN LOOK...

We have now set our filter to see not only our own ZXXXXX data sets, but the data sets starting with ZXP.PUBLIC as well. Nice!

Everyone has the same ZXP.PUBLIC data sets, and you will often use these to copy from. If you get an error that you cannot save or edit something opened from ZXP.PUBLIC, that's because you have read-only access.

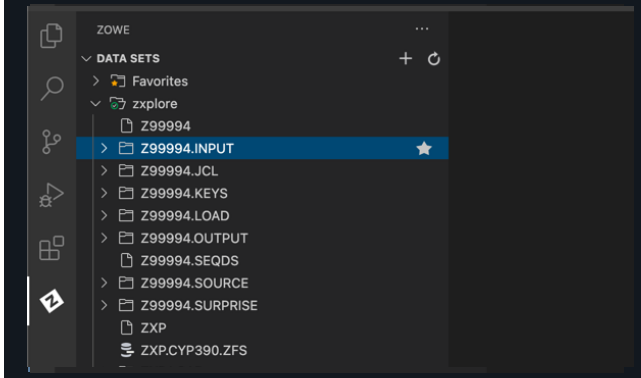
Typically, whenever you have to edit a data set, that will happen within your own ZXXXXX data sets. Just keep that in mind, and don't be surprised when trying to edit a ZXP.PUBLIC data set or member.



3. LET'S MAKE SOME DATA SETS

We have some code written up specifically to build a lot of the data sets and members we'll use for this challenge. You can find that in ZXP.PUBLIC.JCL.

Open that up and look for a member titled PDSBUILD. Right-click on that and select "Submit Job". By the time you're done reading this, all the files required to get started will be in a place, so let's get moving to Step #4.

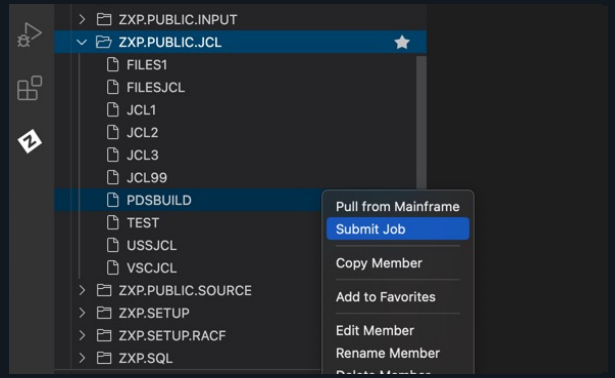


4. SHOW US WHAT YOU GOT

In your “DATA SETS” view, close and reopen your “zxplore”search vew, by clicking on the arrow icons.

Look in your ZXXXXX.INPUT data set. This is a Partitioned Data Set (PDS) and these are members inside it. Look through them, but don’t do anything yet.

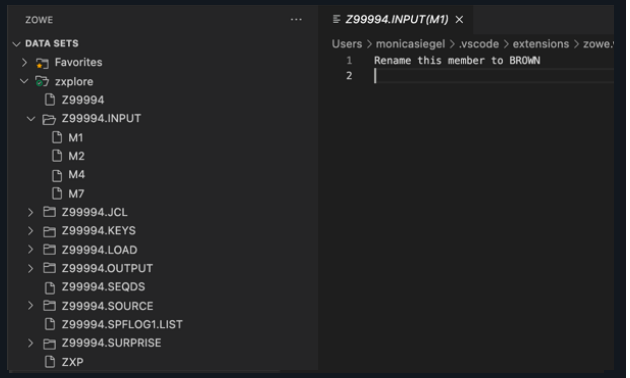
BTW: The set of files you’re looking at were generated just for you, so if you’re working with other people, don’t be alarmed if you get something different than your friends.



5. IF YOU NEED TO START OVER

If at any time you want to reset back to Square 1 for this challenge, just re-run the JCL from Step 3. If you want to be extra-thorough, you can delete the members as well, anything existing in your INPUT data set will be overwritten.

You will get the same “random” files as before, so no worries about having to figure out a whole new set of names.



6. RENAME IN THE MAINFRAME

One of the members in your INPUT data set will contain text directing you to rename it, and the name you should rename it to. You will need to click on the member to see its contents load in the editor on the right side.

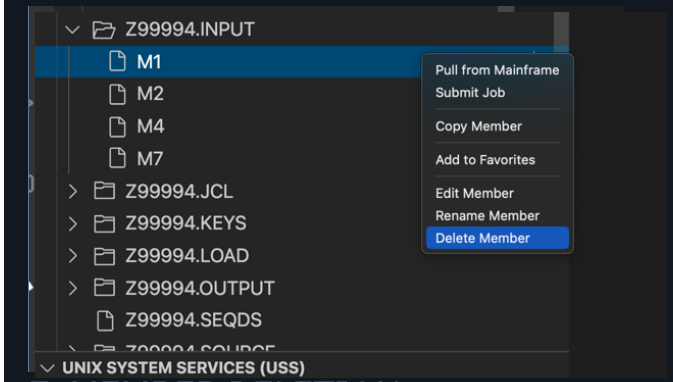
To rename a member, just right-click on it, and select “Rename Member”. Then simply enter the new name in the dialog box that pops up at the top of your VS Code window.

”WHY CAN’T WE JUST HAVE FILES AND FOLDERS AND CALL IT A DAY?”

Different applications access their data differently. Some applications read thousands and thousands of records of customer data one right after the other. They will always process the records from top to bottom, in the order they are given, and for these applications, Sequential Data Sets provide the best format so that information can be processed as fast as possible.

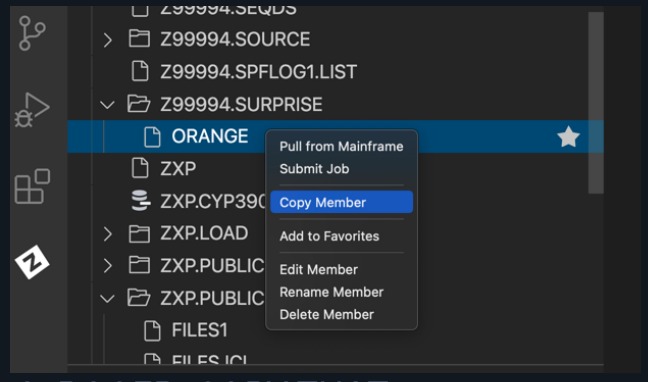
Other times, an application doesn’t know what data will be needed next and having members within a Partitioned Data Set allows the flexibility needed to have everything available in any order it is needed.

There are other data sets you’ll use as well, but for now, we’re just focusing on Partitioned and Sequential.



7. MEMBER DELETION

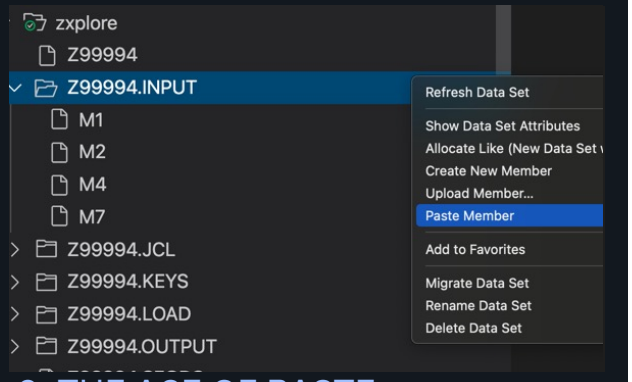
Similarly, track down the member that is directing you to delete it. Right-click on that one and select “Delete Member”. Confirm that you really do want to delete it in the dialog box that pops up, and Poof, it’s gone, just like that!



8. ROGER, COPY THAT

Look in the SURPRISE data set, because *SURPRISE* there’s yet another data set in there for you. Open it up and read what’s inside. It should be asking you to copy it and then paste it into your INPUT data set.

Start by Right-clicking on the source data set (the one you found in SURPRISE) and select “Copy member”. Now you’re all primed and ready to paste it in Step #9.



9. THE ACE OF PASTE

With the source data set member safely copied, right click on your INPUT data set and select “Paste Member”. It will prompt you for a name. This is a chance for you to move the contents of a file from one place to another with a new, unique name, but in this case, we want to keep the same name, so just call it what it originally was in the SURPRISE data set (hint: it should be a color, so you’ll wind up with two data set members with the names of colors).

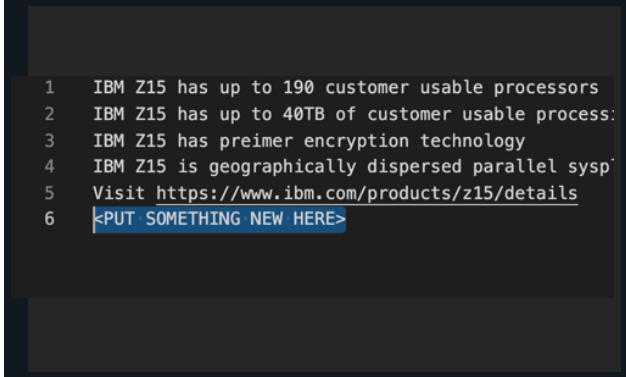
“THIS ALL SEEMS KIND OF SIMPLE. AM I MISSING SOMETHING?”

Data sets and members aren’t the trickiest thing in the world, but there are a few peculiarities that some people get confused by, especially for those who are new to working on computers. Modern touchscreen interfaces hide away a lot of the complexities that professionals need to be aware of.

For example, try right-clicking on a data set and select “Show Data Set Attributes”.

In addition to there being Sequential and Partitioned data sets, data sets have a number of attributes which can be set to obtain the best performance, security, and scalability. We take the default values for this challenge, but a true Z Systems Programmer will know the best values for all of these fields.

Just something to keep in mind when working your way through these challenges.



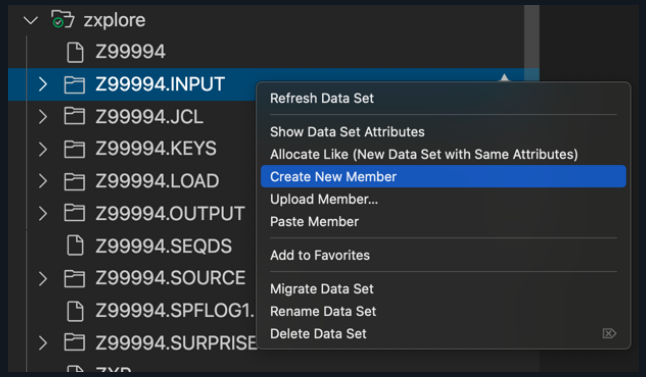
10. SEQUENTIAL SUCCESS

We've been using our INPUT Partitioned Data Set (PDS) to hold the members for this challenge. Now let's look at the SEQDS data set located at ZXXXXX.SEQDS.

(*substitute your own ID for ZXXXXX)*

Instead of separate members, this data set contains its own records. This is known as a Sequential Data Set. When you open it up, you'll see that each record is represented as a line of the data set. Adding a new record is as simple as adding a new line and entering text.

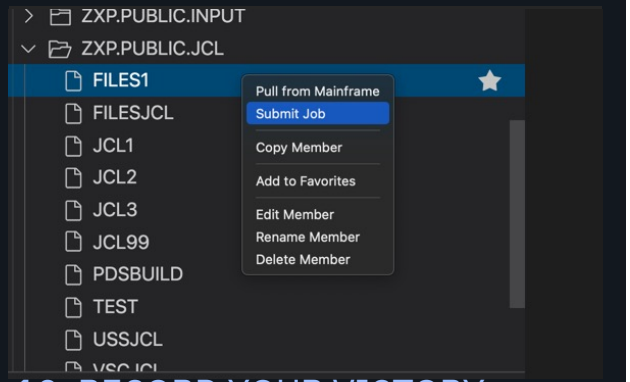
For this step, we want you to do just that; add a new line (record) and say hello, tell us how the weather is, what your favorite cartoon is...anything you like (as long as it's appropriate), then save the file and you're done with this step.



11. MAKE YOUR OWN MEMBER

Right-click on your INPUT data set and select "Create New Member". Give it the name of MYNEWMEM when prompted for a name.

Creating a new member within a data set is as simple as that. If you get an error, make sure you're not adding any additional spaces or punctuation, as member names can only be 8 characters or less.



12. RECORD YOUR VICTORY

By now, you should have FIVE members in your INPUT data set (Two that were there originally, one that you renamed, one you created, and another that you copied from SURPRISE). It should NOT contain the member which we asked you to delete. Your SEQDS data set should also have an additional record (line) with your input. Double check your work, then find ZXP.PUBLIC.JCL(FILES1). Right click on it and select "Submit Job" to hand in your work.

NICE JOB! LET'S RECAP

You're getting really hand with those data sets and records. Just keep in mind that there are many different ways of storing files, and part of your job is to figure out whether the situation calls for a Partitioned Data Set, a Sequential Data Set, a Member within a Partitioned Data Set, or something else.

Properly optimized data is what keeps applications running fast.

NEXT UP...

It's one thing to follow steps, it's another to understand. In the next two challenges, we'll get to the bottom of exactly what these sets and members are, plus dive a bit deeper into JCL.You're making great progress, let's keep it all going.