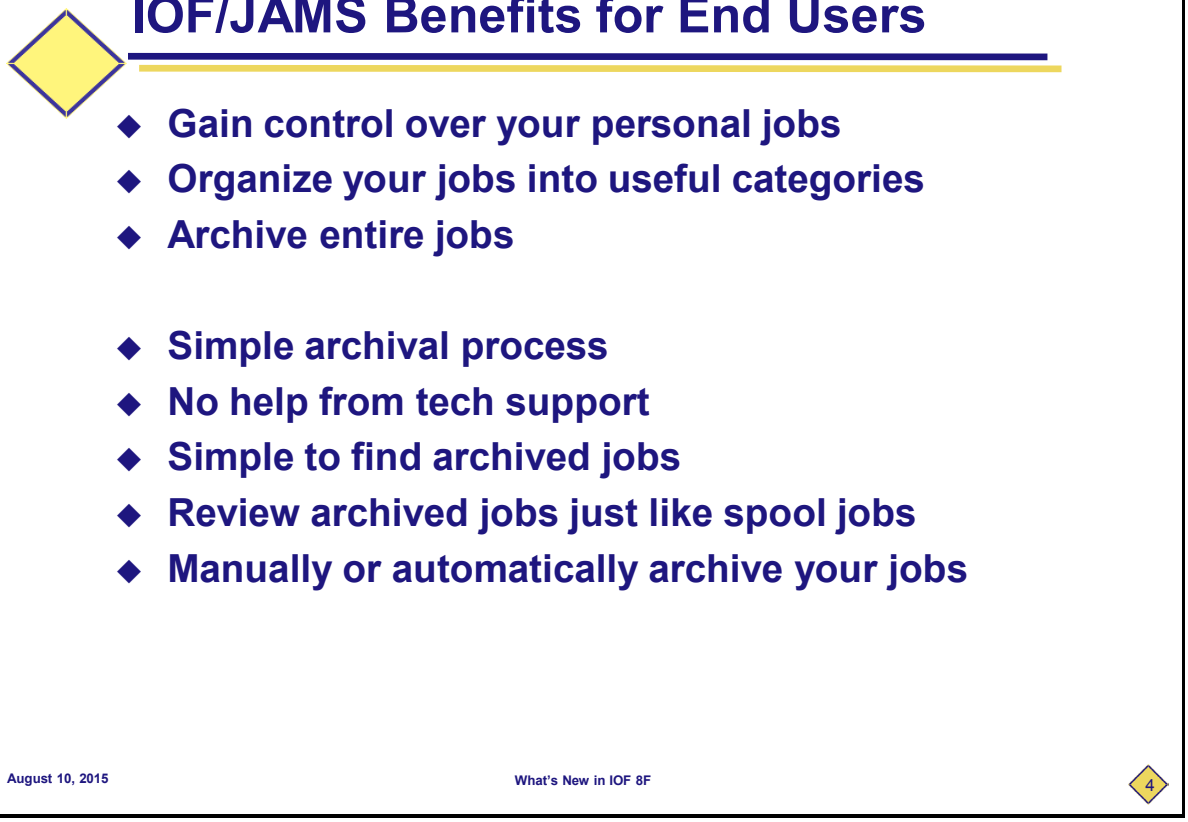
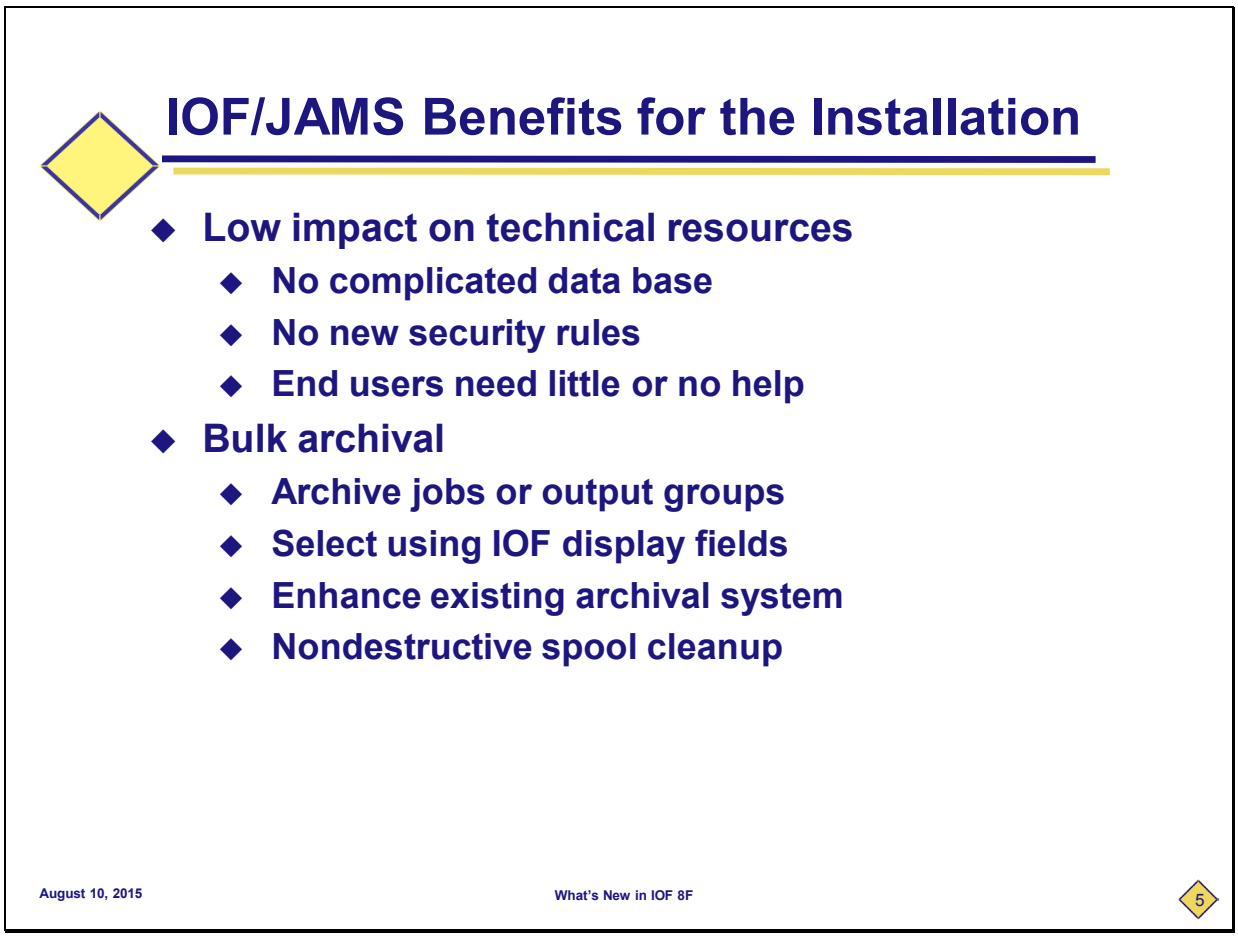
IOF/JAMS

- Standard z/OS data sets
- No complex data base
- No new security rules

So, what is IOF/JAMS? It's a very simple archival system that archives each job to a standard z/OS data set. That means that you don't need a complex data base or new security rules. If you can access a job with IOF, you can archive it. If you can create z/OS data sets, you can create archive data sets.



OK, this sounds pretty simple, but why would you want to use it?
First of all, you gain complete control over your jobs because they are in your data sets. And, you can organize your jobs into categories that can be very useful. We'll show you how we use that in IOF development. We also archive the entire job, including SYSIN data sets and the entire original input job.
This may sound pretty interesting at this point, but you're probably wondering how complicated it is to use. I think you will see that users can easily learn to archive their jobs with no help at all from tech support. And, it's also very simple for them to find and browse archived jobs.
Best of all, users review their archived jobs just like their spool jobs. And, they can manually archive jobs from IOF displays or automatically archive jobs after jobs run.

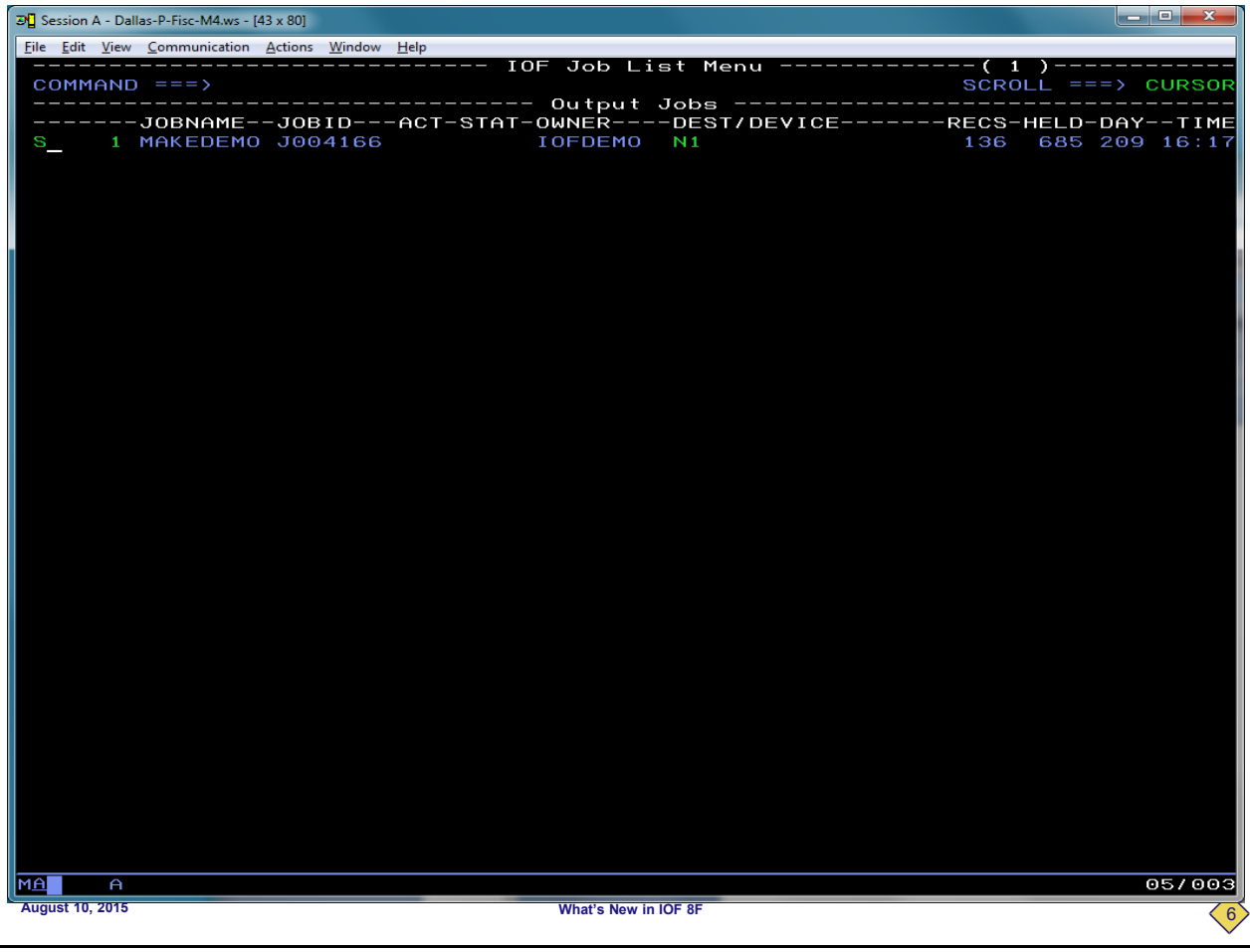


Here are some benefits that we see for the installation. First, users can start archiving with little or no help from tech support. That's because there are no complicated data bases or new security rules.

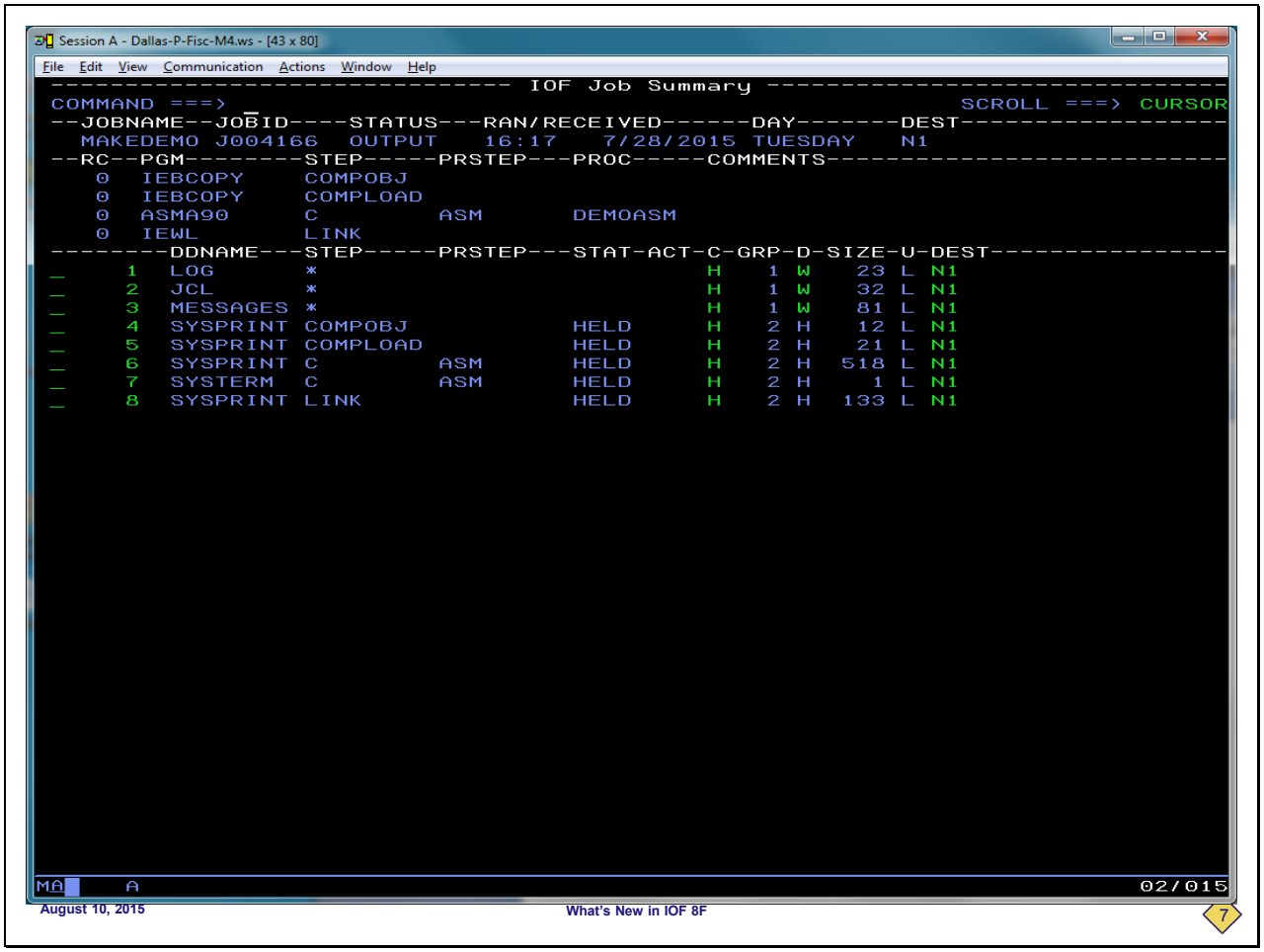
IOF/JAMS also includes a classical bulk archival capability that lurks around and picks up jobs based on very flexible selection criteria. For IOF/JAMS, you can select based on any IOF display fields. Many companies already have a bulk archival system in place, but here are a couple of potentially useful applications even if you already have an archive system.

You can actually use IOF/JAMS to archive additional information for jobs that are already being archived (without disrupting their normal archival). As we will see, IOF/JAMS saves an IOF Job Summary for each job and the original input job. You can choose to have IOF/JAMS just archive that additional information and not any large reports.

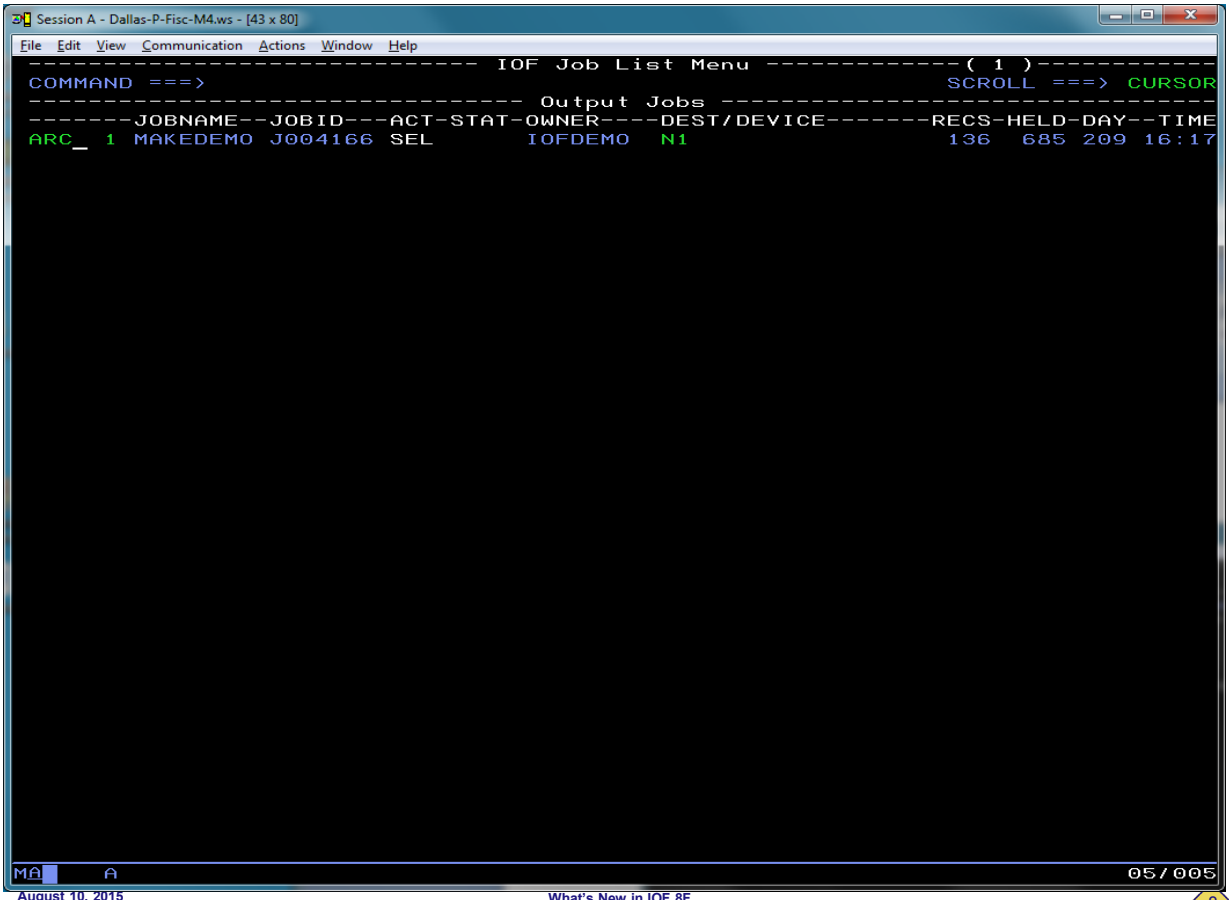
IOF/JAMS can also be very useful for spool cleanup. The potential downside to accidentally throwing away vital jobs can be overwhelming. IOF/JAMS allows you to copy jobs off the spool into z/OS data sets that can then be migrated off the system. And, it's easy to go back and find them.



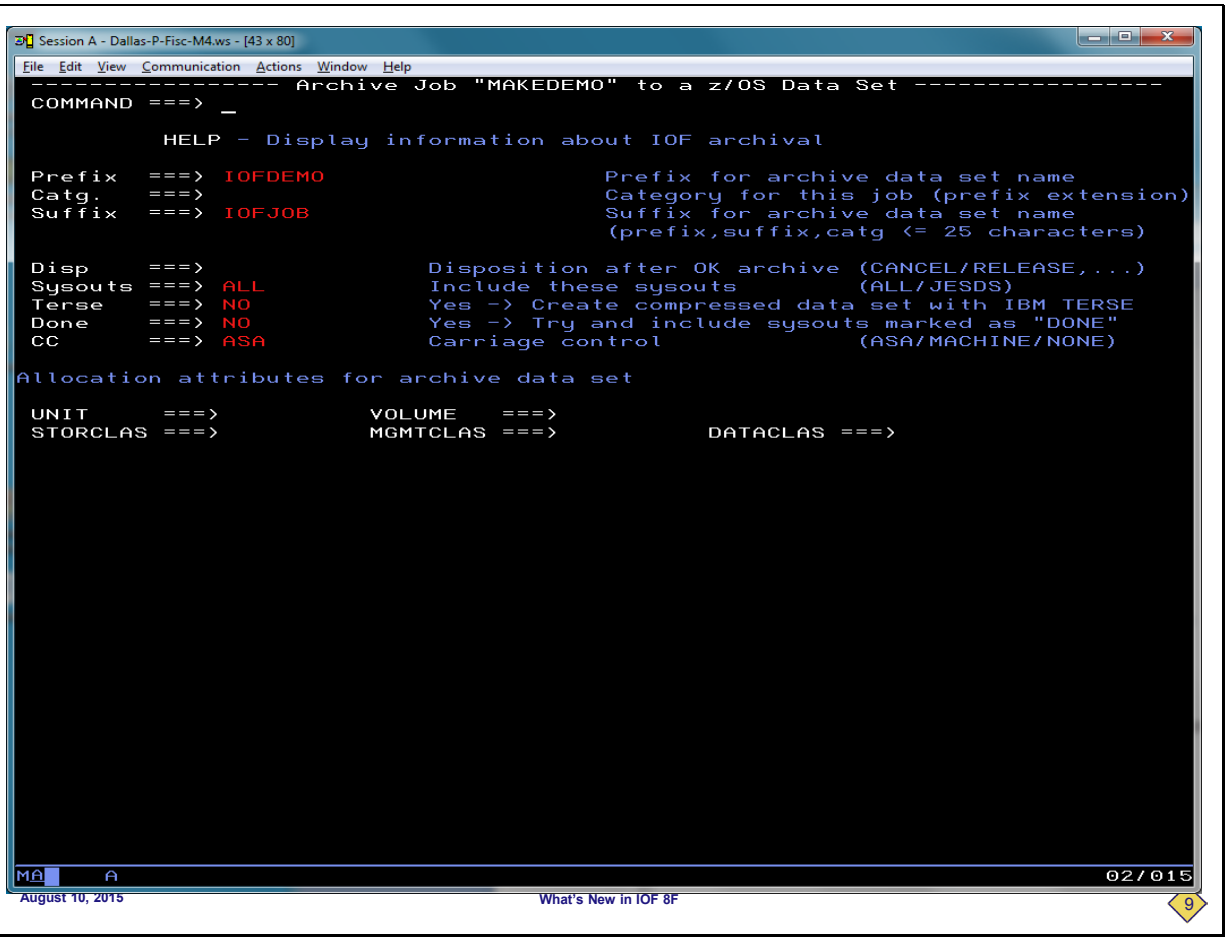
Now, we will demonstrate IOF/JAMS with some screen captures from an actual IOF session. First, we will demonstrate how to manually archive jobs.
This is a standard *IOF Job List Menu*. For simplicity of this demo, this job list has only one job.
To display the normal *IOF Job Summary* panel for this job, we will enter the "S" line command by the job.



This is the *IOF Job Summary* panel for the MAKEDEMO job. This is the IOF display that shows you the return code results for the jobs and all of its sysout data sets. We are displaying it here because we will see the archived version of it after we archive the job. We will return to the *Job List Menu*.

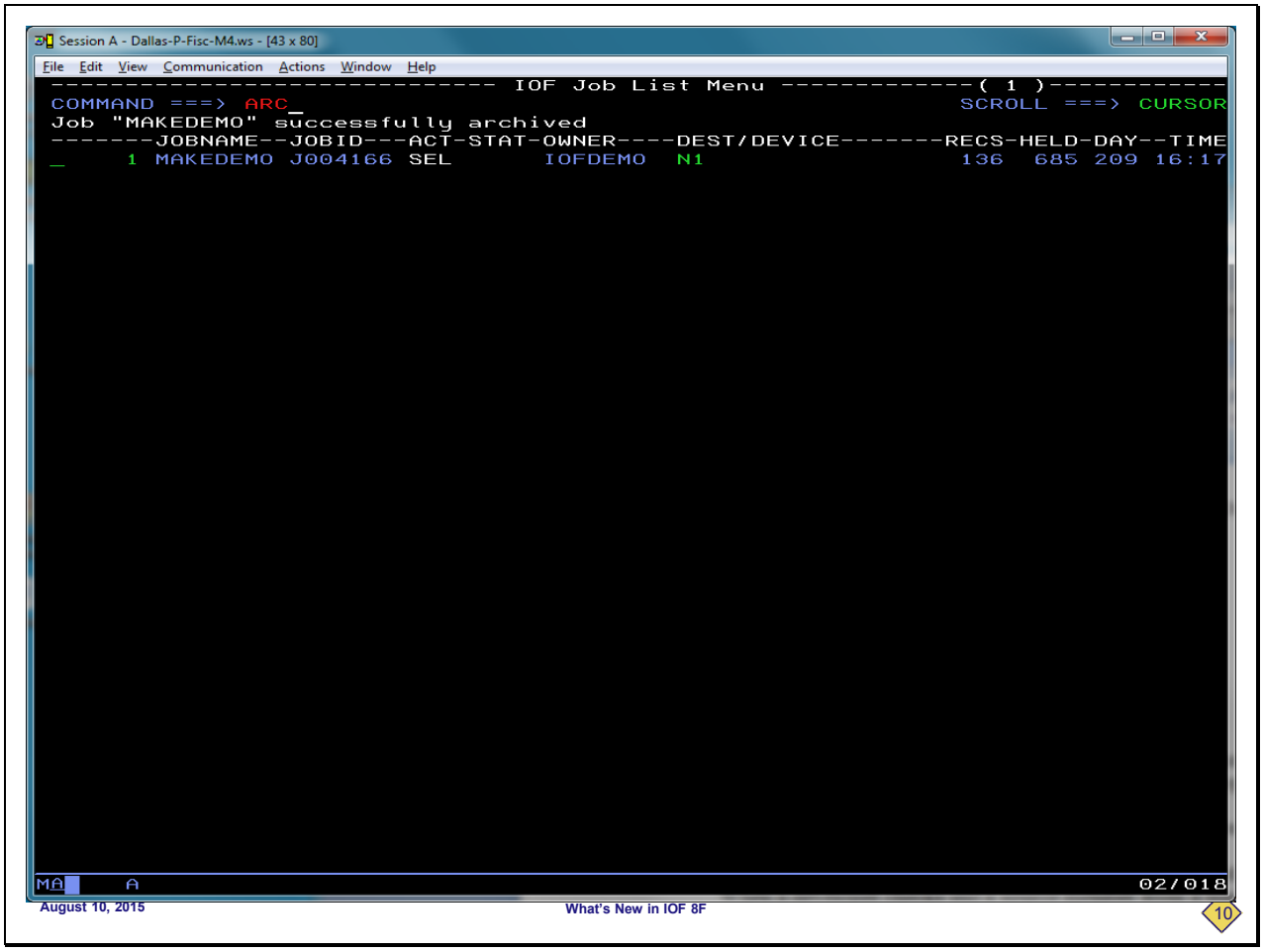


We will now enter the ARC line command for the MAKEDEMO job to manually archive the job.

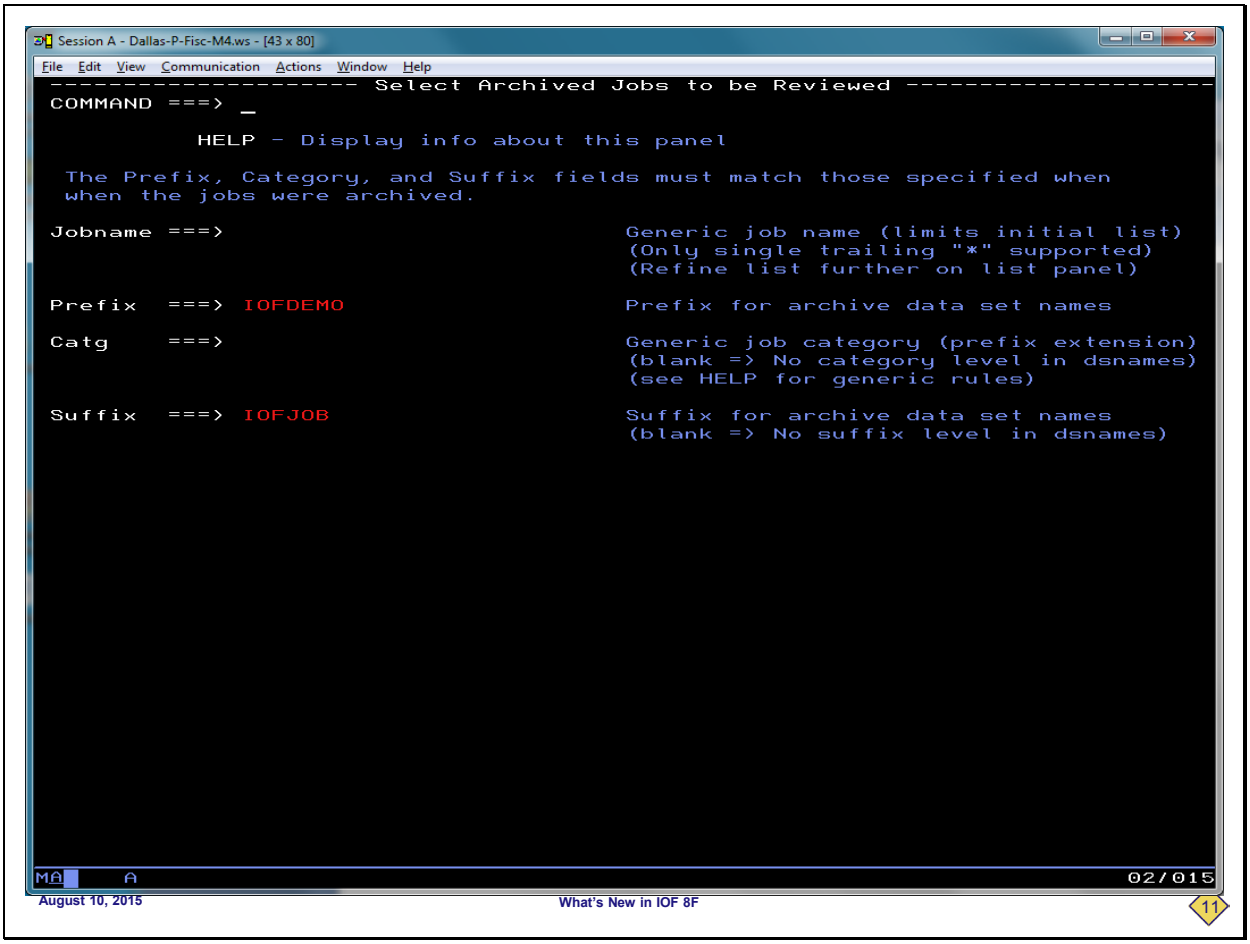


This is where you specify the target data set for the archival and other useful options. Since we have previously archived other jobs, it has remembered the data set name prefix and suffix that we like to use for our archive data sets

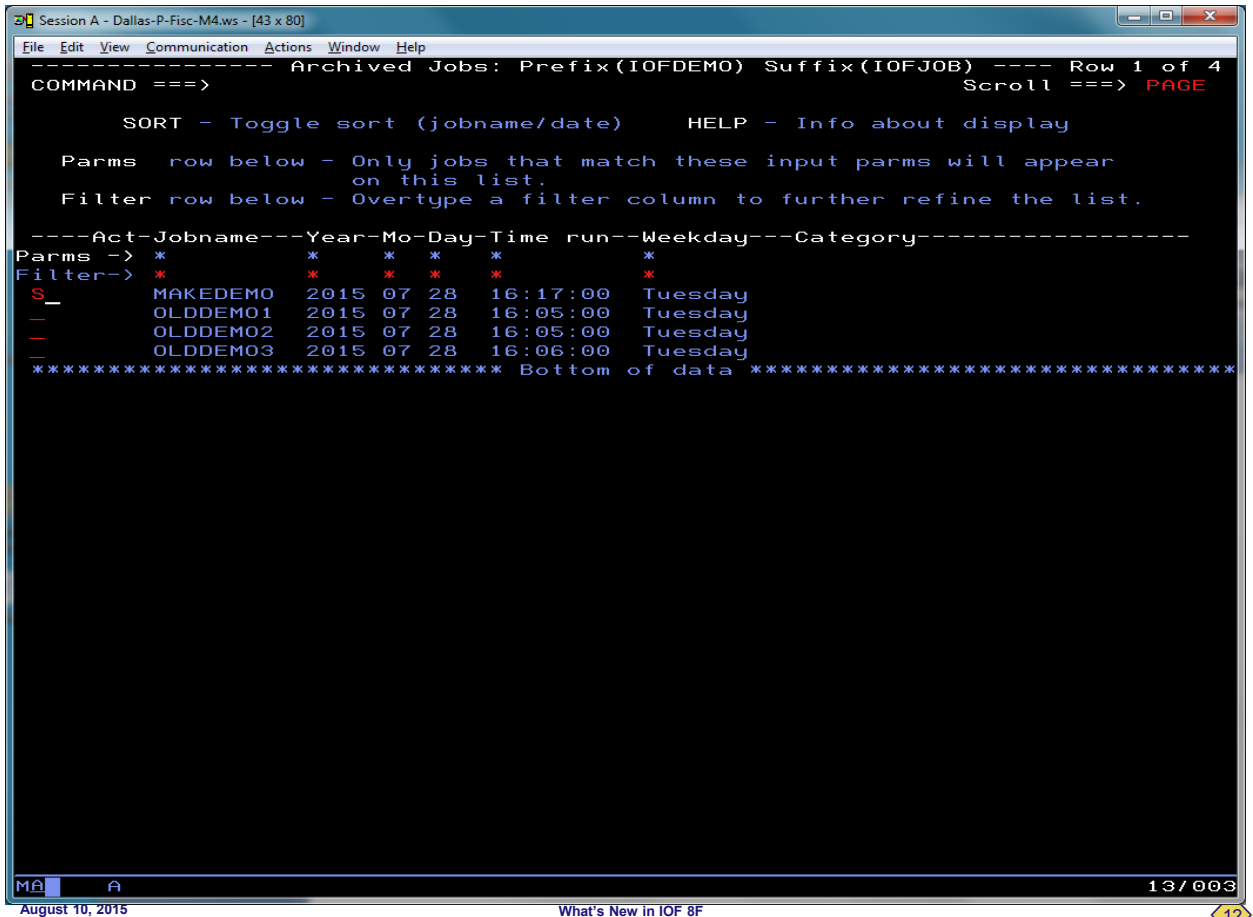
You optionally can specify a second level for the data set name in the category field. We will show how useful this can be later in the presentation. For now we will ignore the fields at the bottom of the panel that show its flexibility and press ENTER to archive the job.



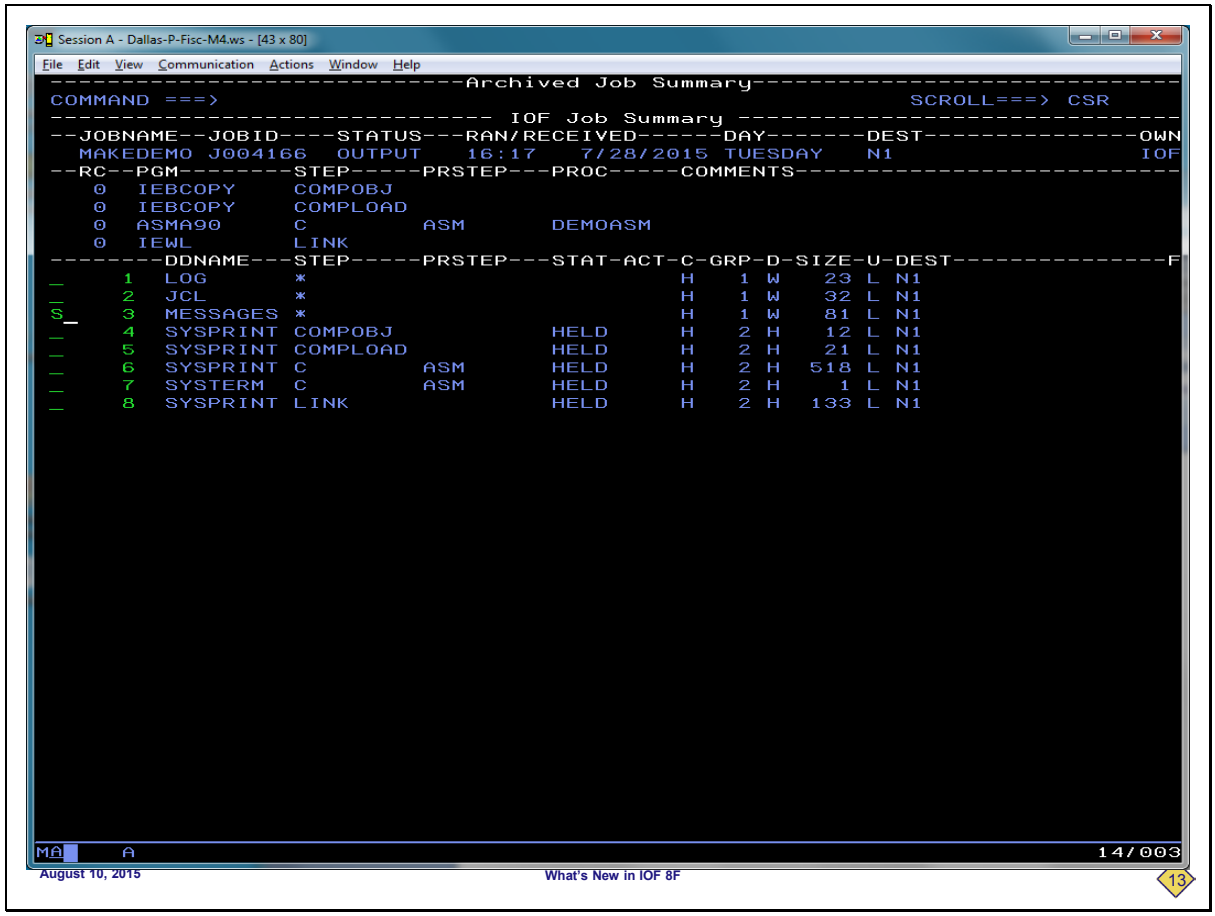
We can see that the message indicates that we have archived the job. We will now enter the ARC primary command to display our archived jobs.



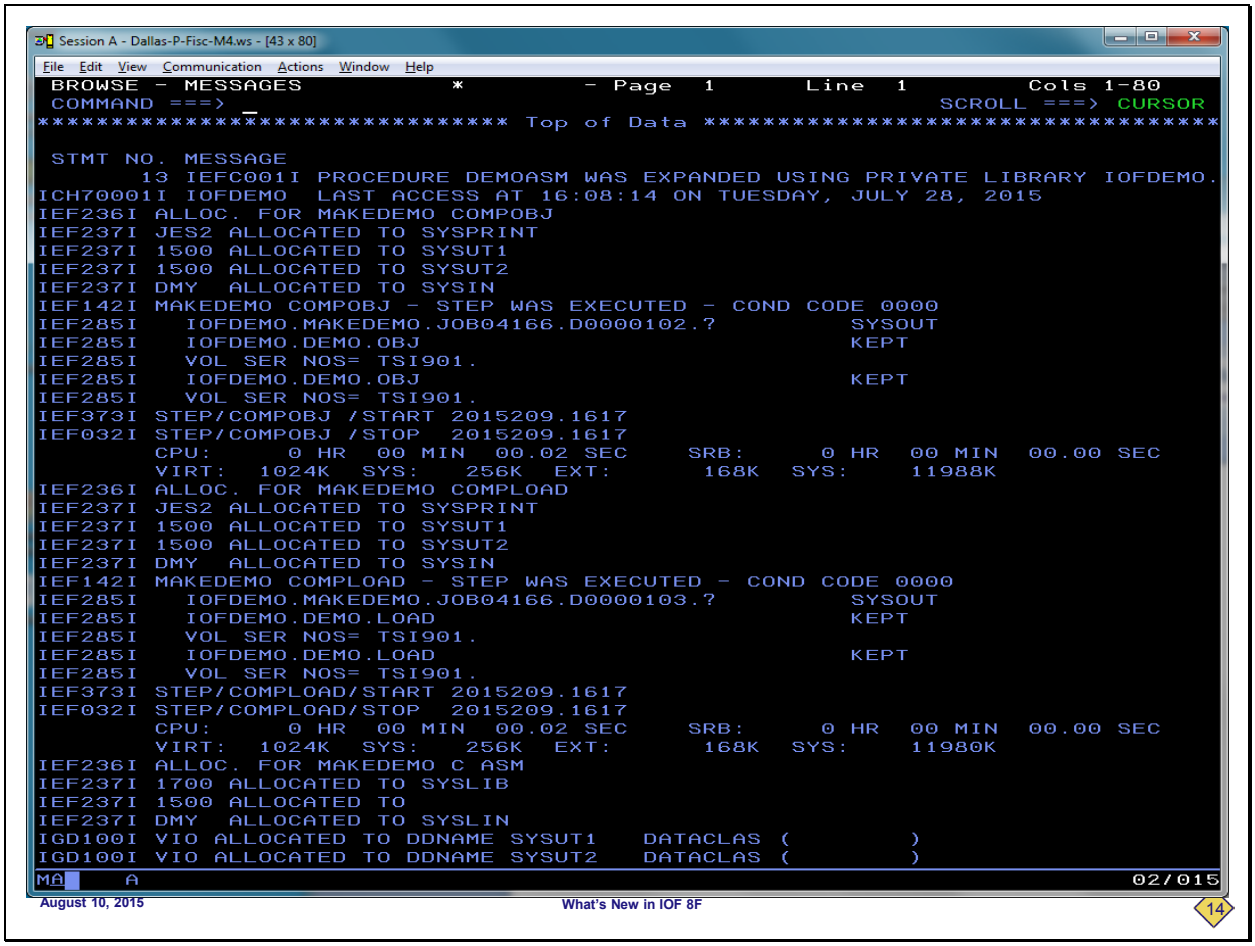
This panel is where you ask for a list of archived jobs. Since it remembers the prefix and suffix from our last archive, we will simply press ENTER to see our archived jobs.



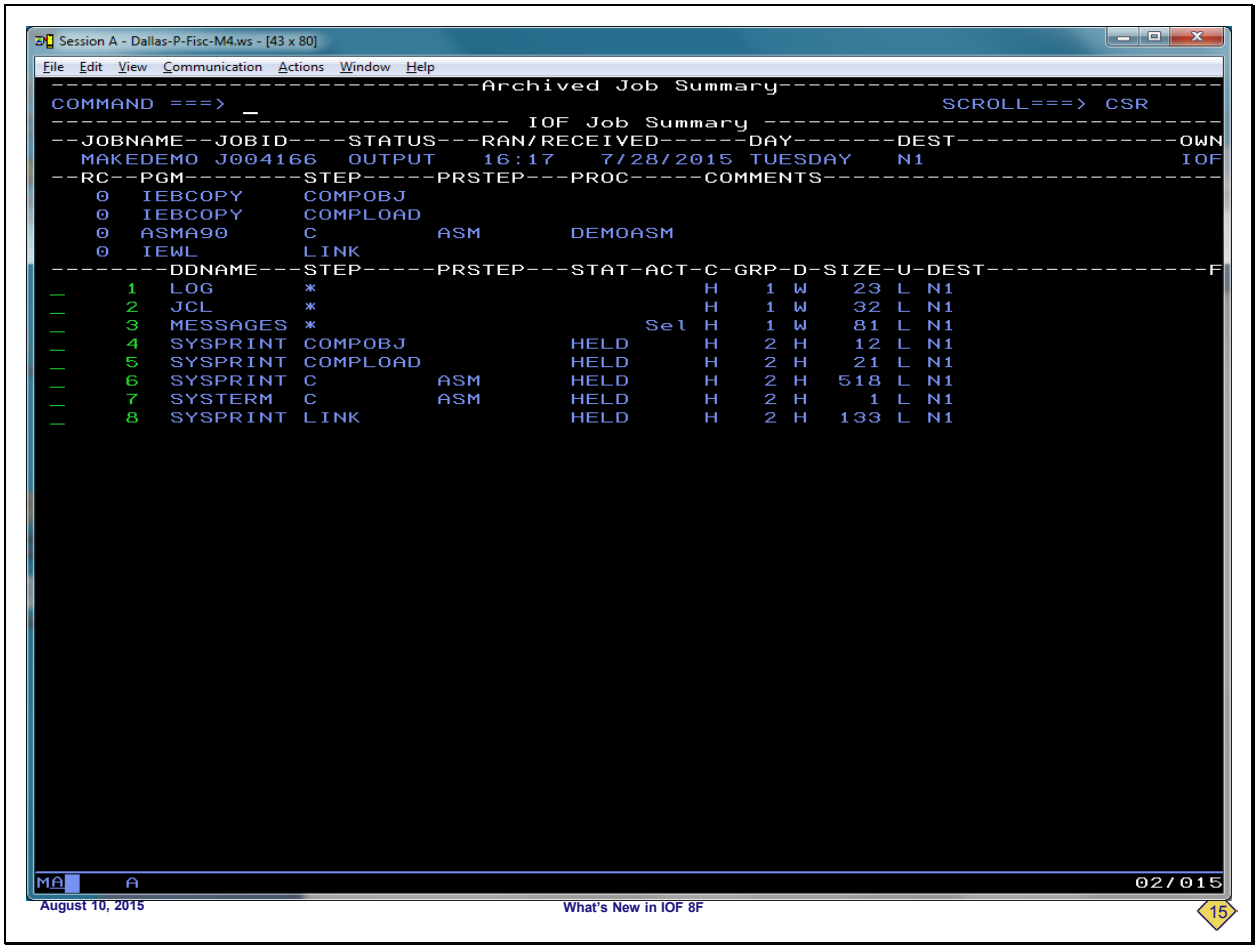
This is a list of jobs that have been archived with the specified data set name prefix and suffix. This list is simple, but it could have hundreds of jobs. The filtering fields can be very useful in refining the list. We can see our MAKEDEMO job, so we will select it.



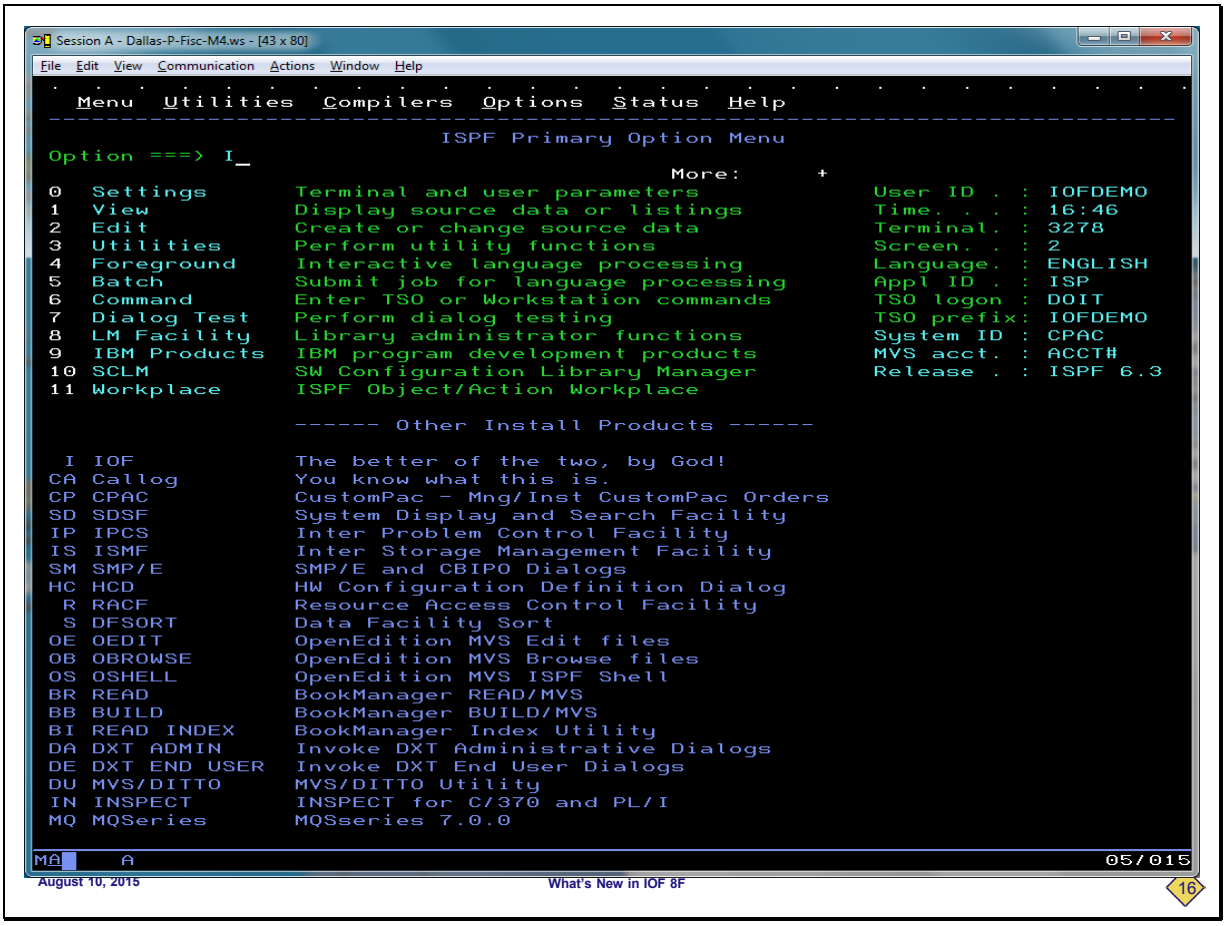
This is the *Archived Job Summary* for the archived job. We are now displaying the data from the archived data set, and from not the spool. We'll see in a minute just how close this is to a normal *IOF Job Summary*. From here we can select a sysout for browse just as you would from the normal *IOF Job Summary* panel.



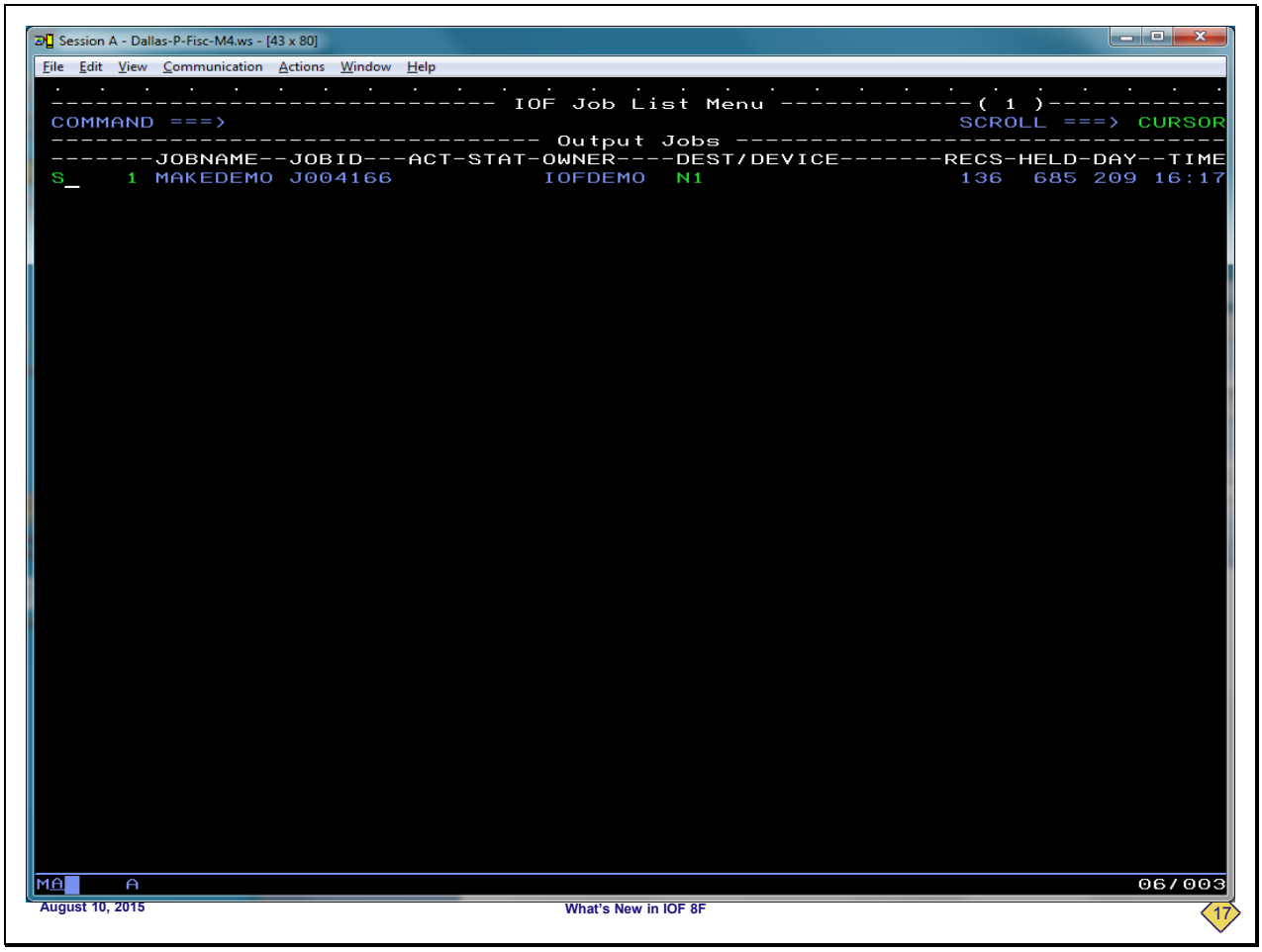
We are browsing this sysout data set from the archived job, not from the spool.
We will return now to the *Archive Job Summary*.



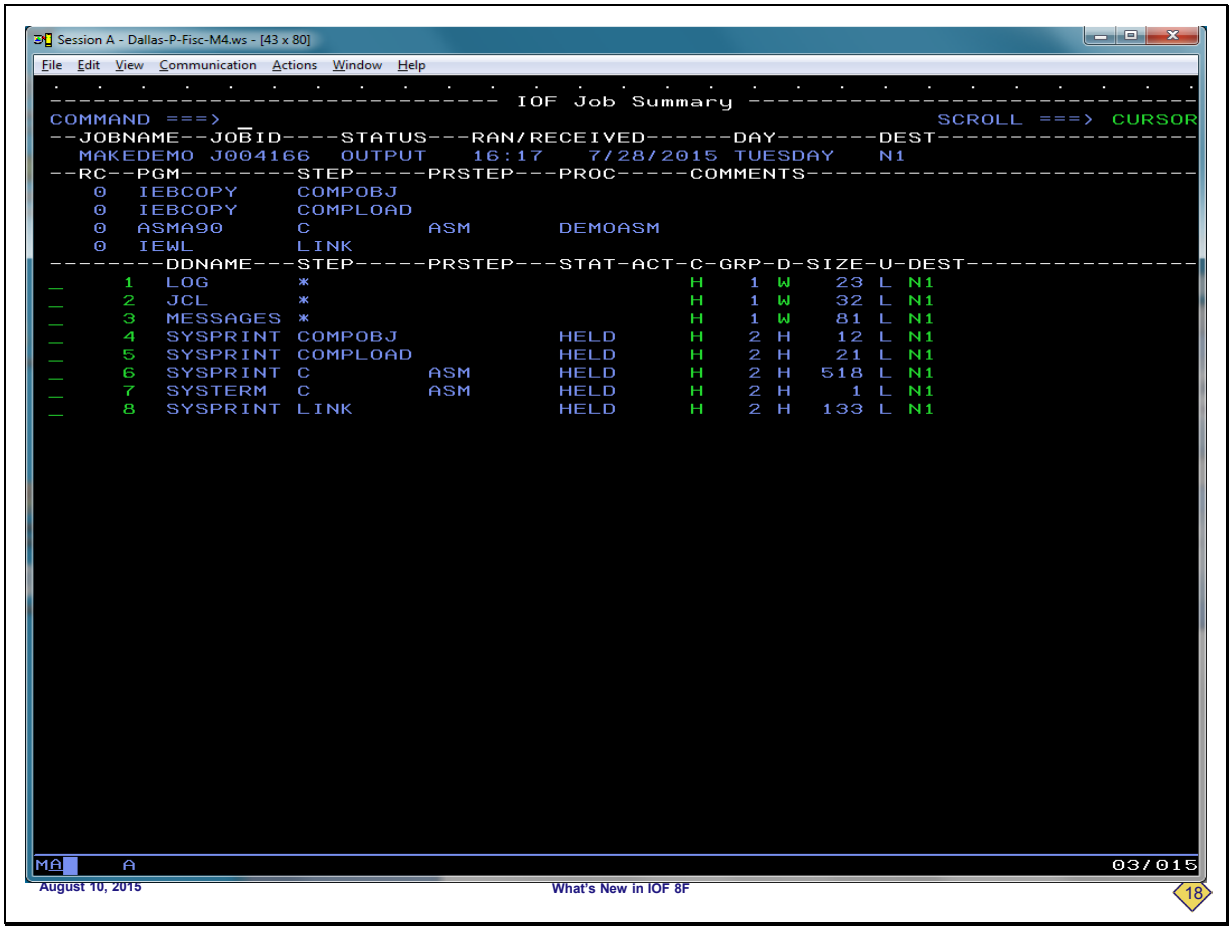
To compare the *Archived Job Summary* for this job with its normal *IOF Job Summary*, we will enter split screen mode.



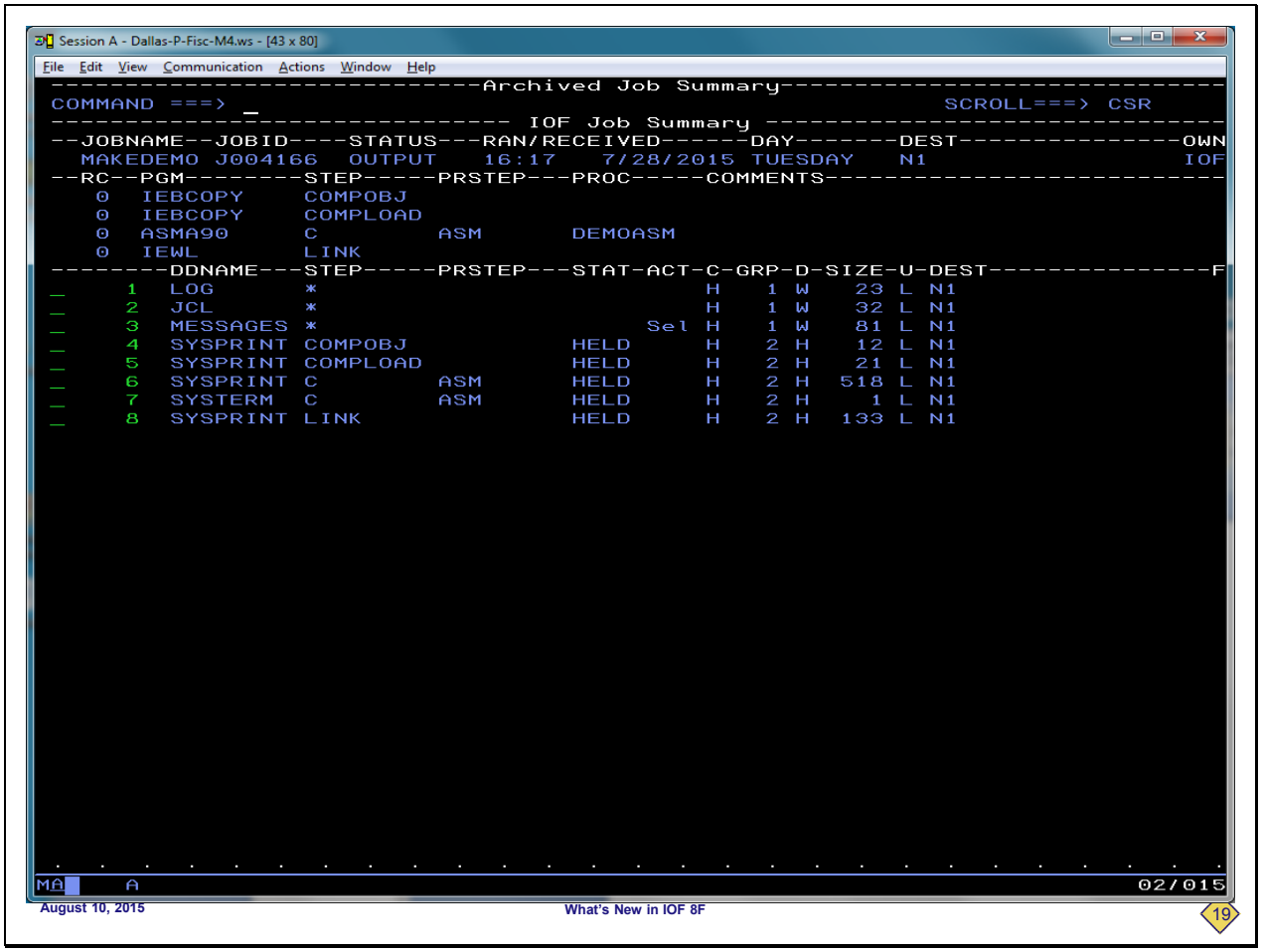
We will enter “I” for IOF and press ENTER.



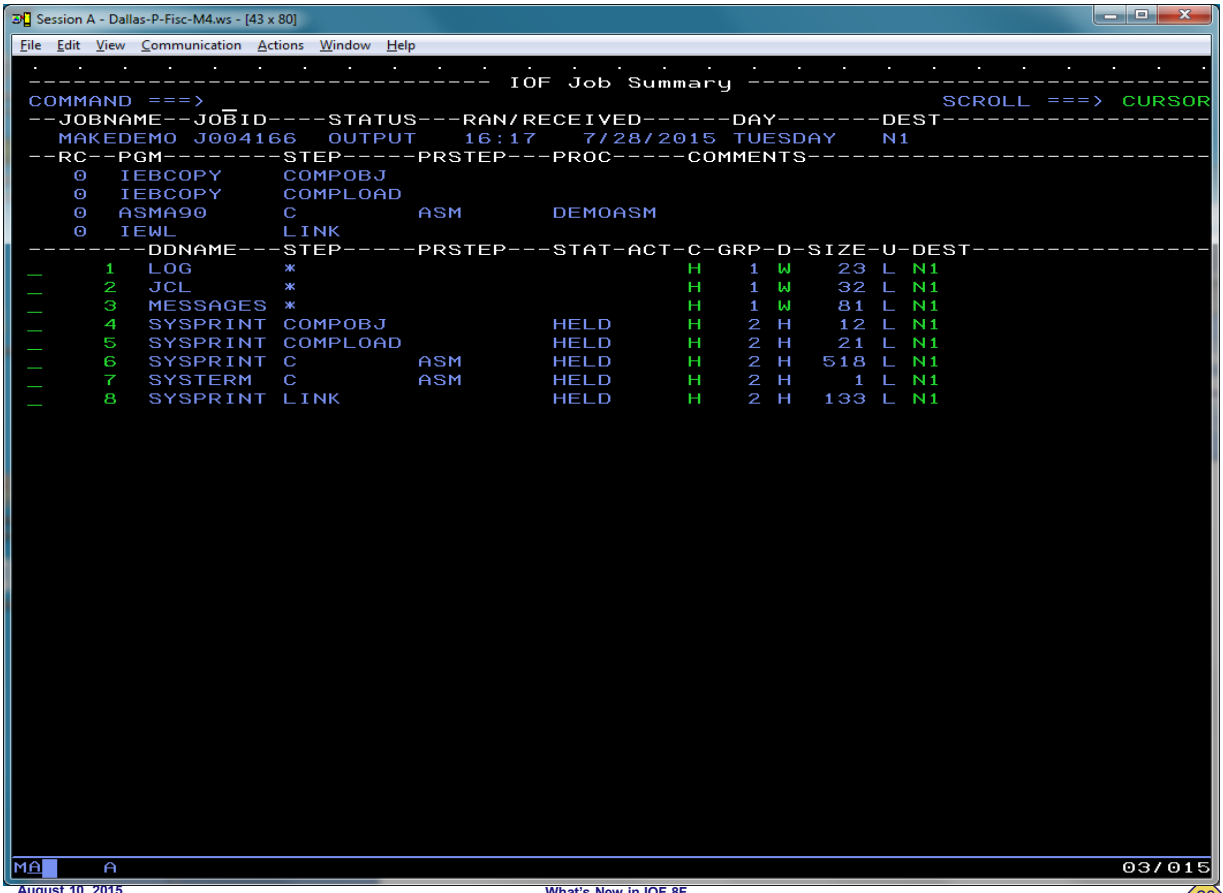
We are now displaying the *IOF Job List*. We will use the "S" line command to redisplay the normal *IOF Job Summary* for the MAKEDEMO job.



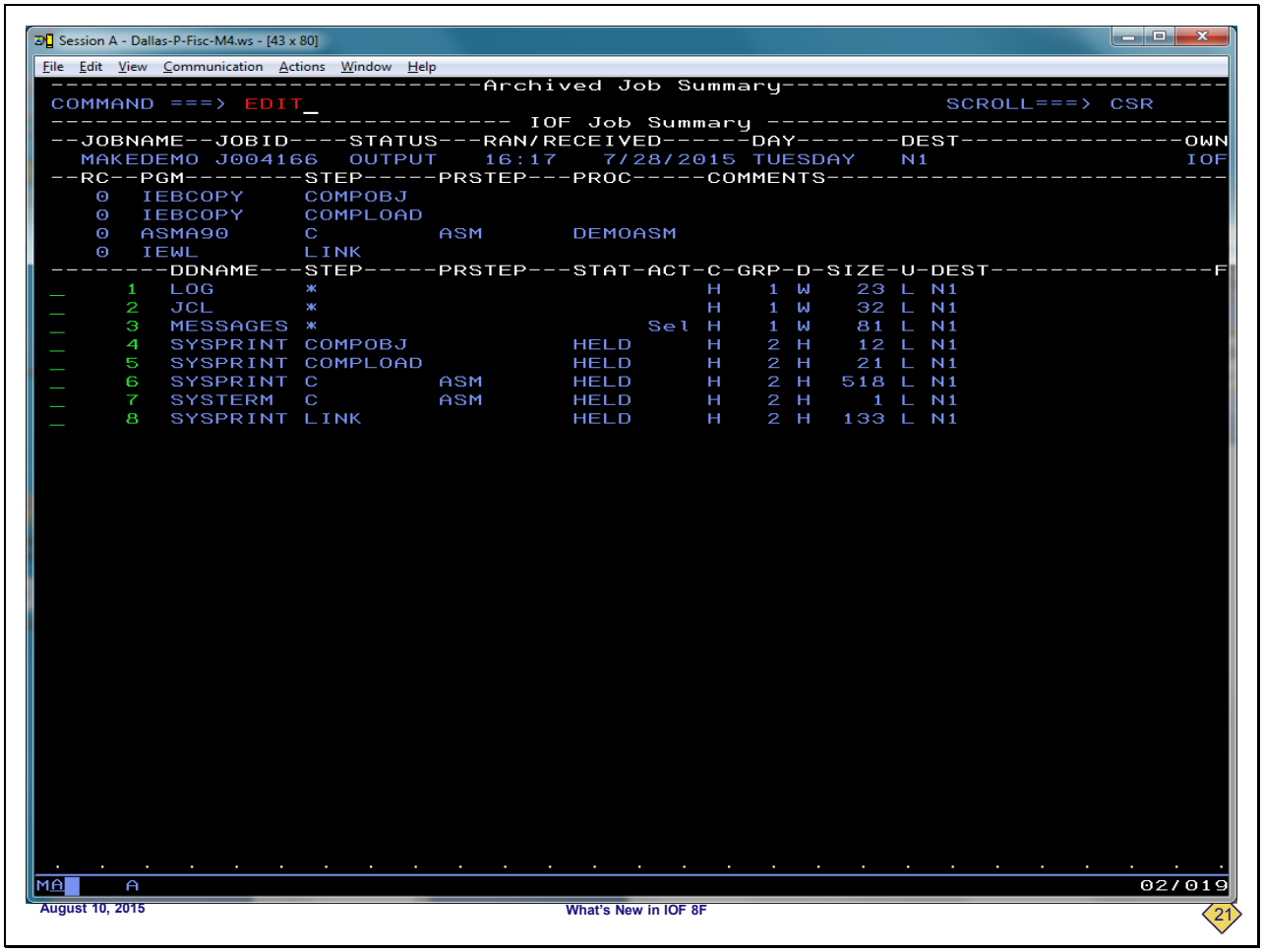
This is the normal IOF Job Summary for the MAKEDEMO job. We will now SWAP back to the Archived Job Summary to demonstrate that reviewing an archived job is exactly like reviewing the original job on the spool.



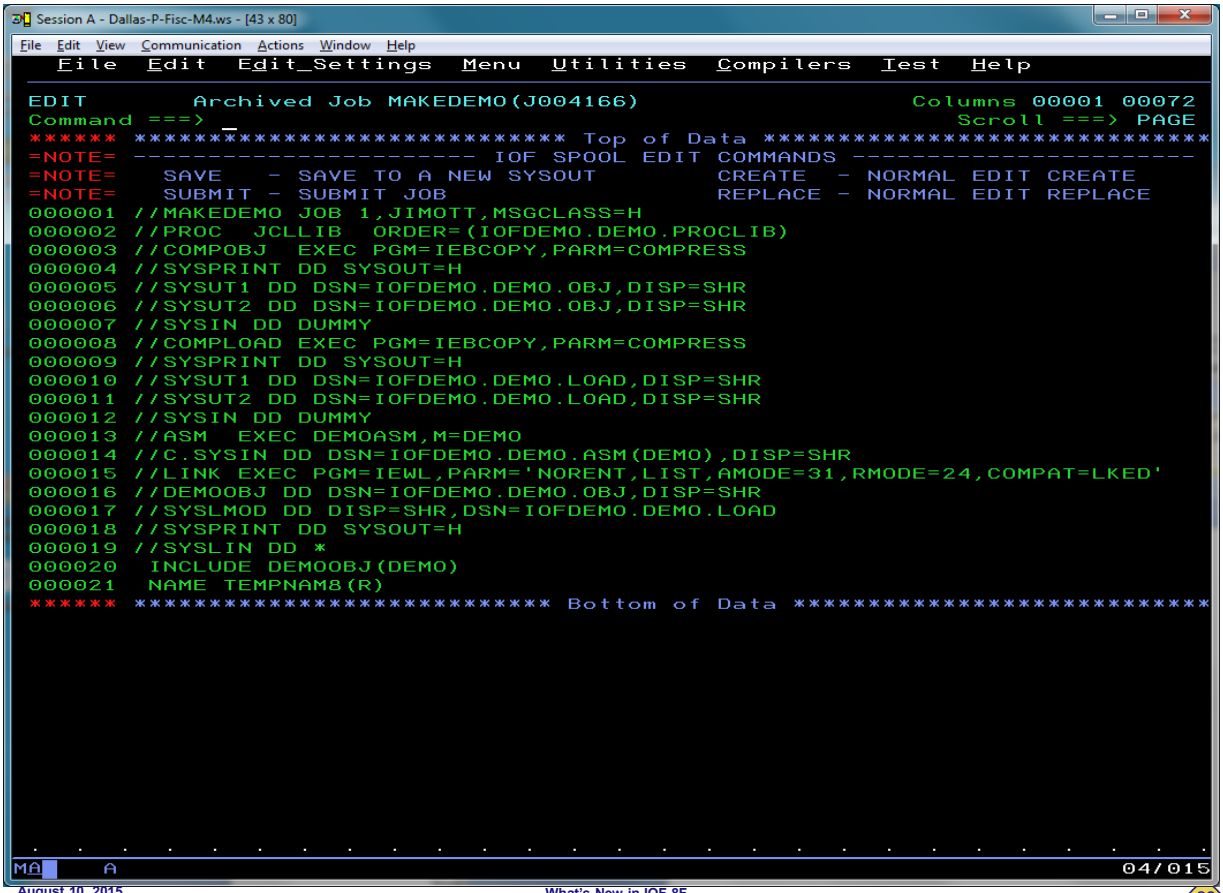
Notice that the *Archived Job Summary* is virtually identical to the original *IOF Job Summary* for the job. This means that you don't need to learn any new tricks to review your archived jobs because they look just like your normal spool jobs. We will enter SWAP again to take another look at the original *IOF Job Summary* panel for the job.



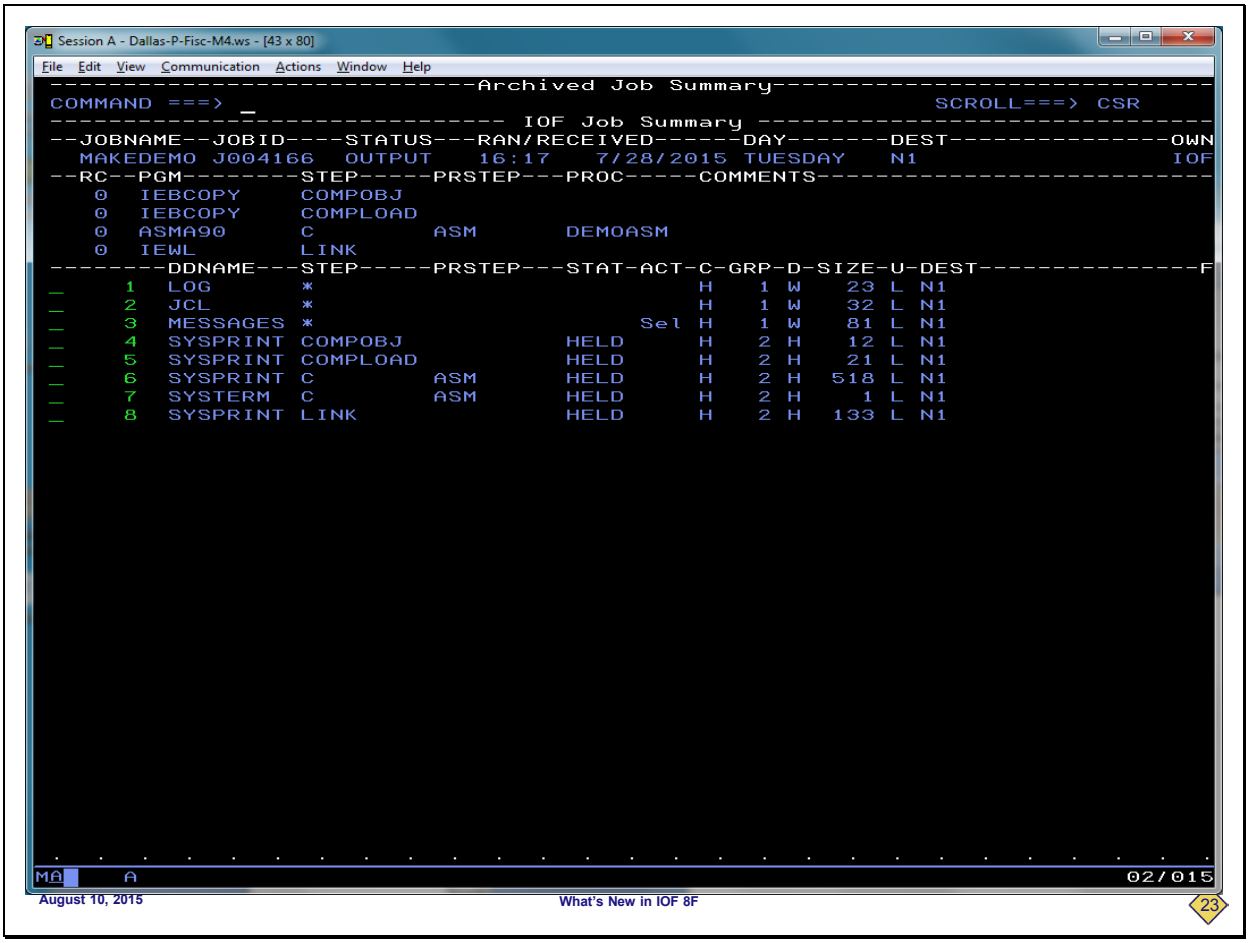
As you can see, the displays are virtually identical.
One more SWAP will take us back to the *Archived Job Summary*.



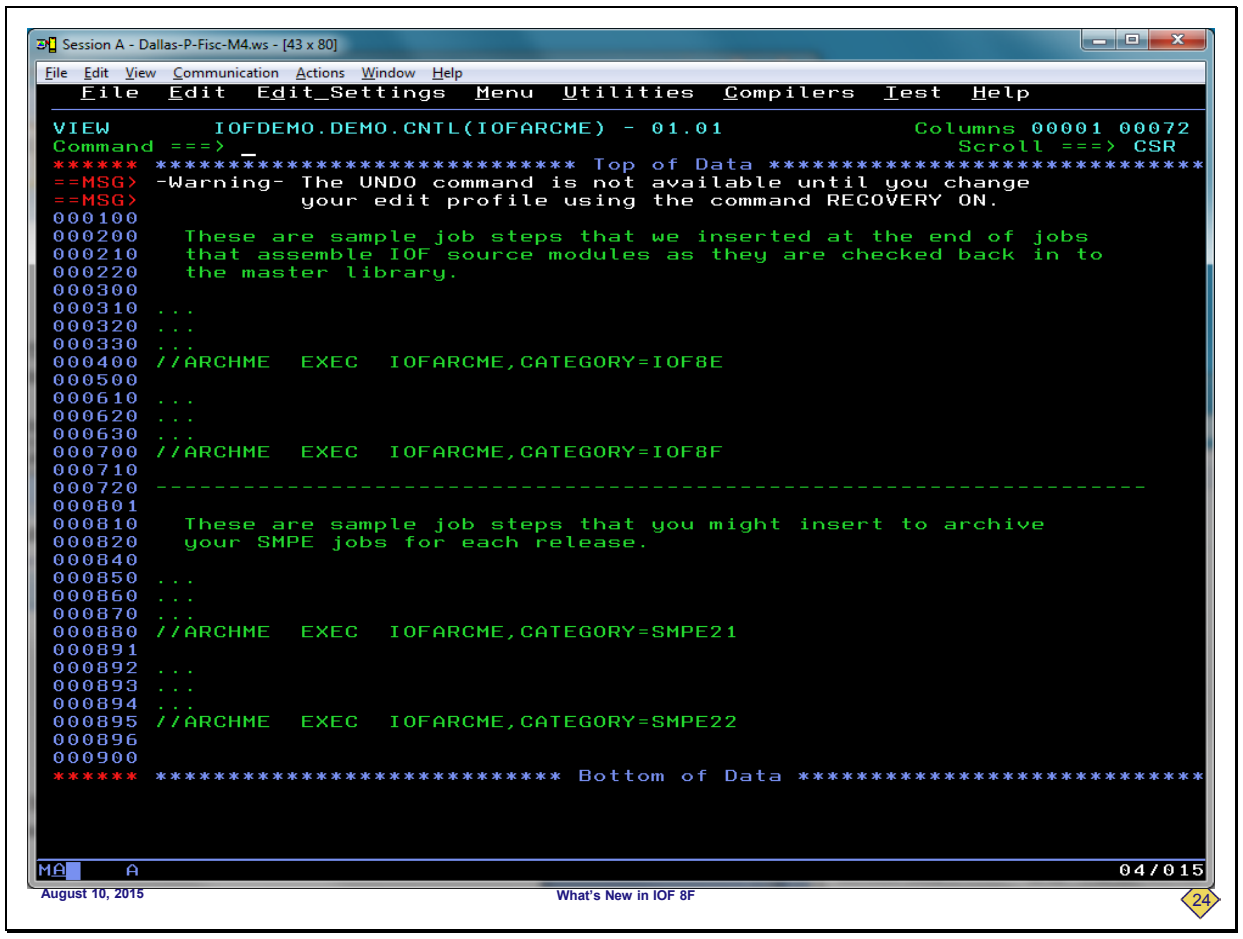
Here we are again on the *Archive Job Summary*. To demonstrate that we have archived the entire original input job, we will enter the EDIT primary command just as we would on a normal *IOF Job Summary* panel.



As you can see, we are now in ISPF edit for the original input job. Think how useful it can be to go back and see the actual input job for an archived job.
We will just return to the *Archive Job Summary*.



I hope we have been able to show you just how simple it is to manually archive jobs and review them with IOF/JAMS.



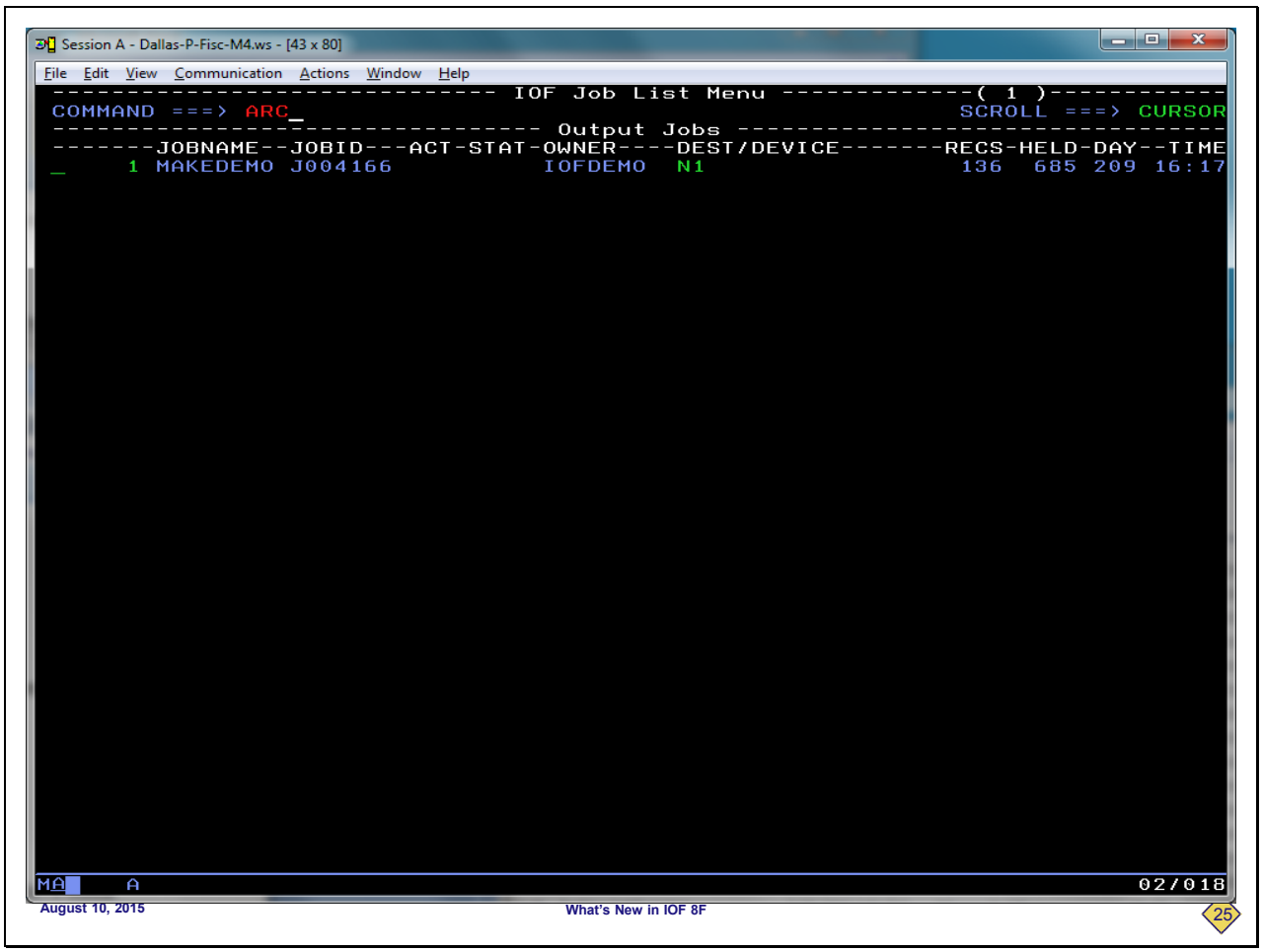
However, in practice, many users may prefer to archive their jobs automatically. The best way to illustrate this is to show you how we use this capability in IOF development. Since we are archiving to z/OS data sets, there's no reason that a job can't archive itself by adding a simple archive step at the end of the job. This screen shows some simple archival steps that we have used in IOF development.

When a developer checks out a source member from a master library, he may compile it many times as it is changed and tested. But when the member is checked back into the master library, a special compile is done that represents the new master source member. That is accomplished with generated batch jobs, and we added a new step to the end of those jobs similar to the ones shown here.

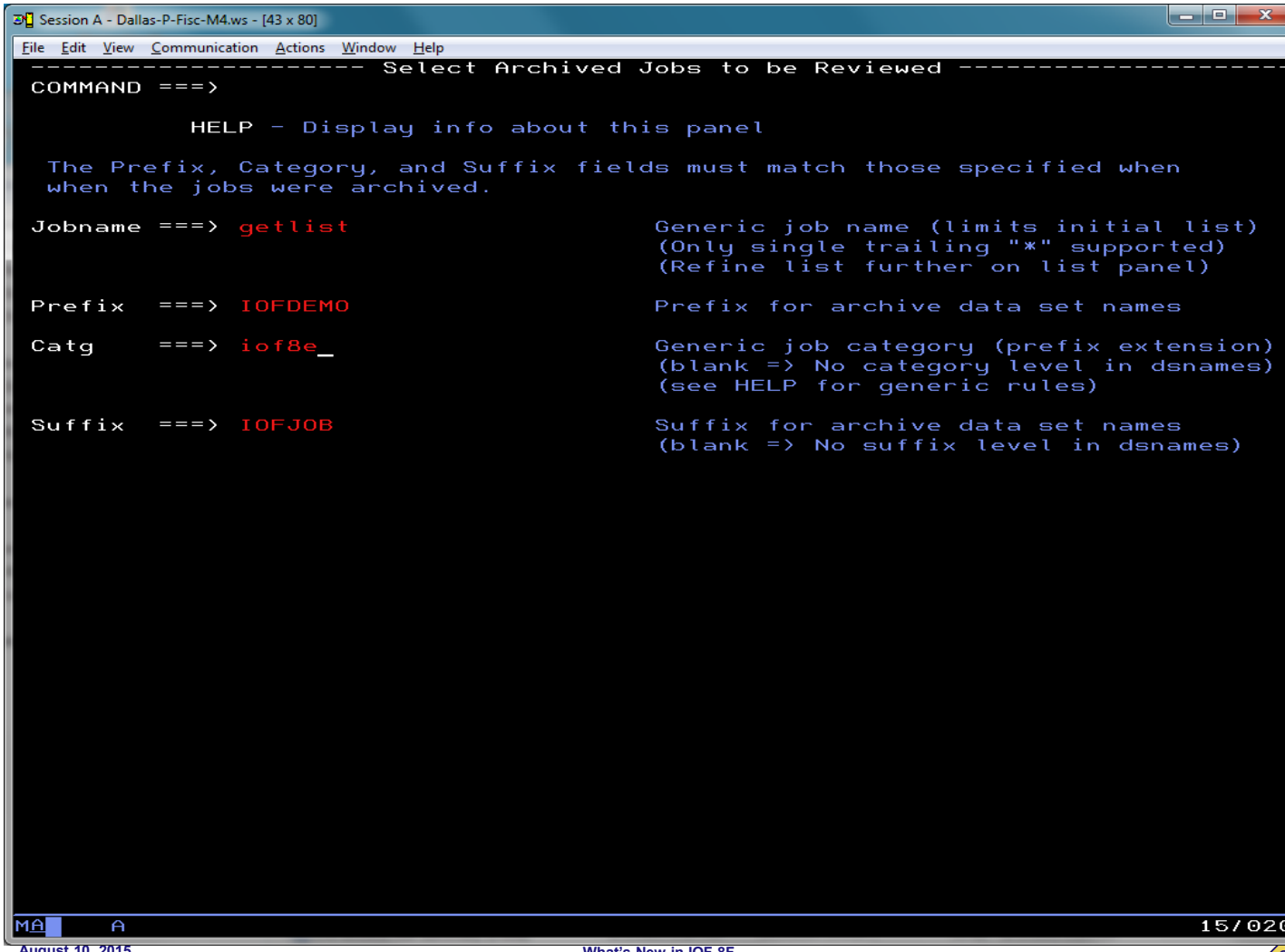
The first example was used during development of IOF Release 8E. The second example was used during the development of 8F. We will show how useful this can be.

The examples at the bottom demonstrate how you might use a similar scheme to organize your various system build jobs.

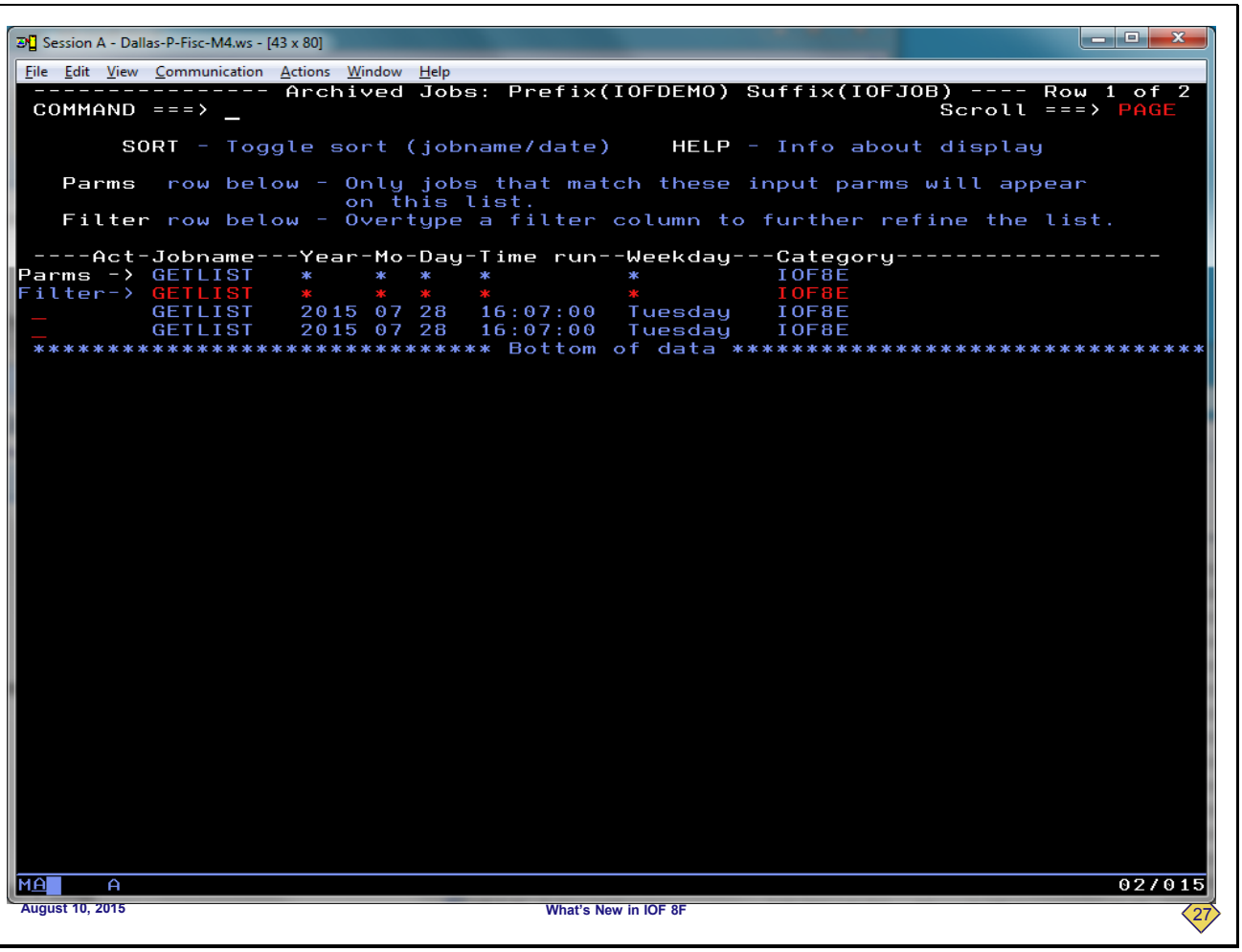
Now, we will take a look at how handy it can be to have these jobs organized.



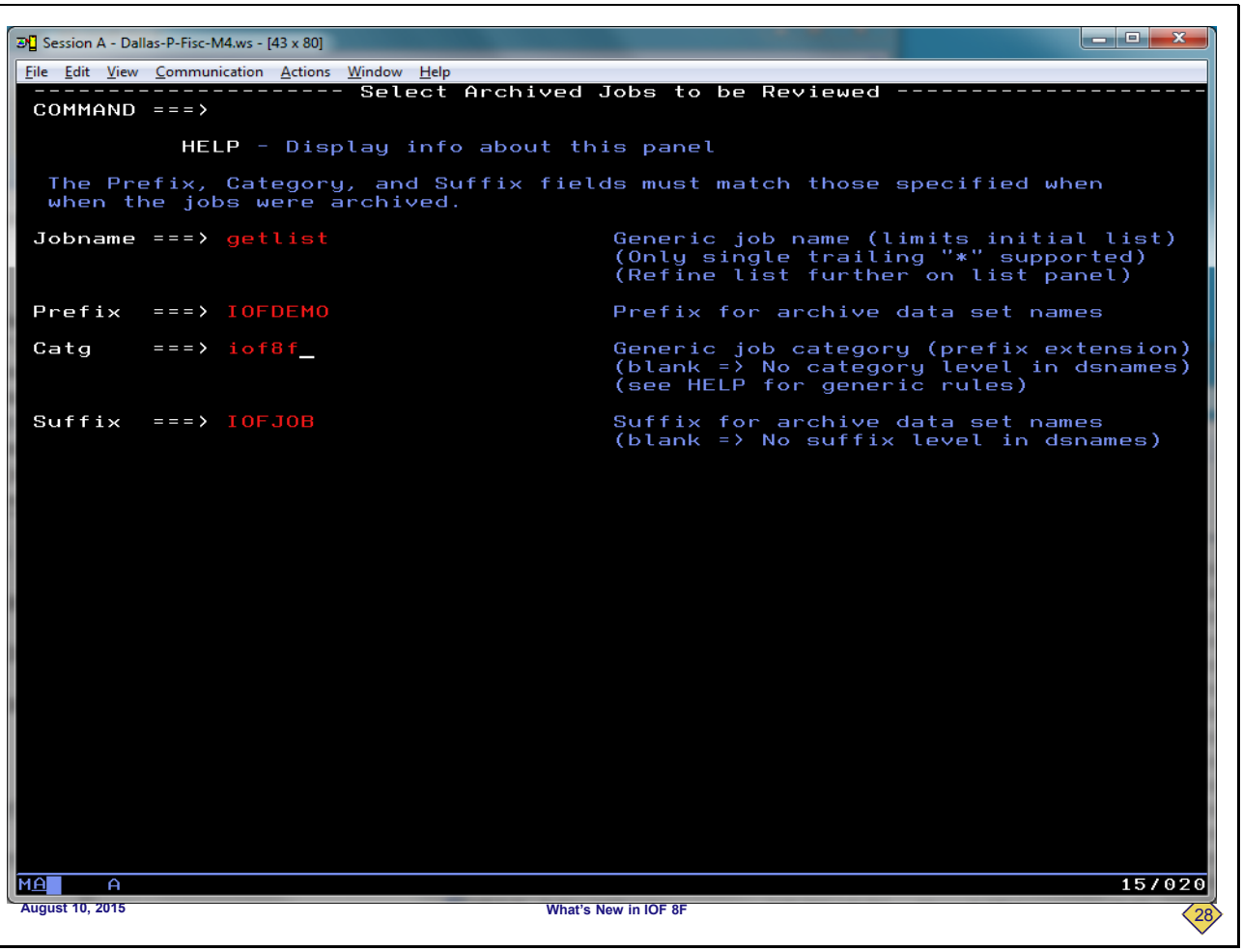
This is a normal *IOF Job List*. We will enter the ARC primary command again to look at archived jobs.



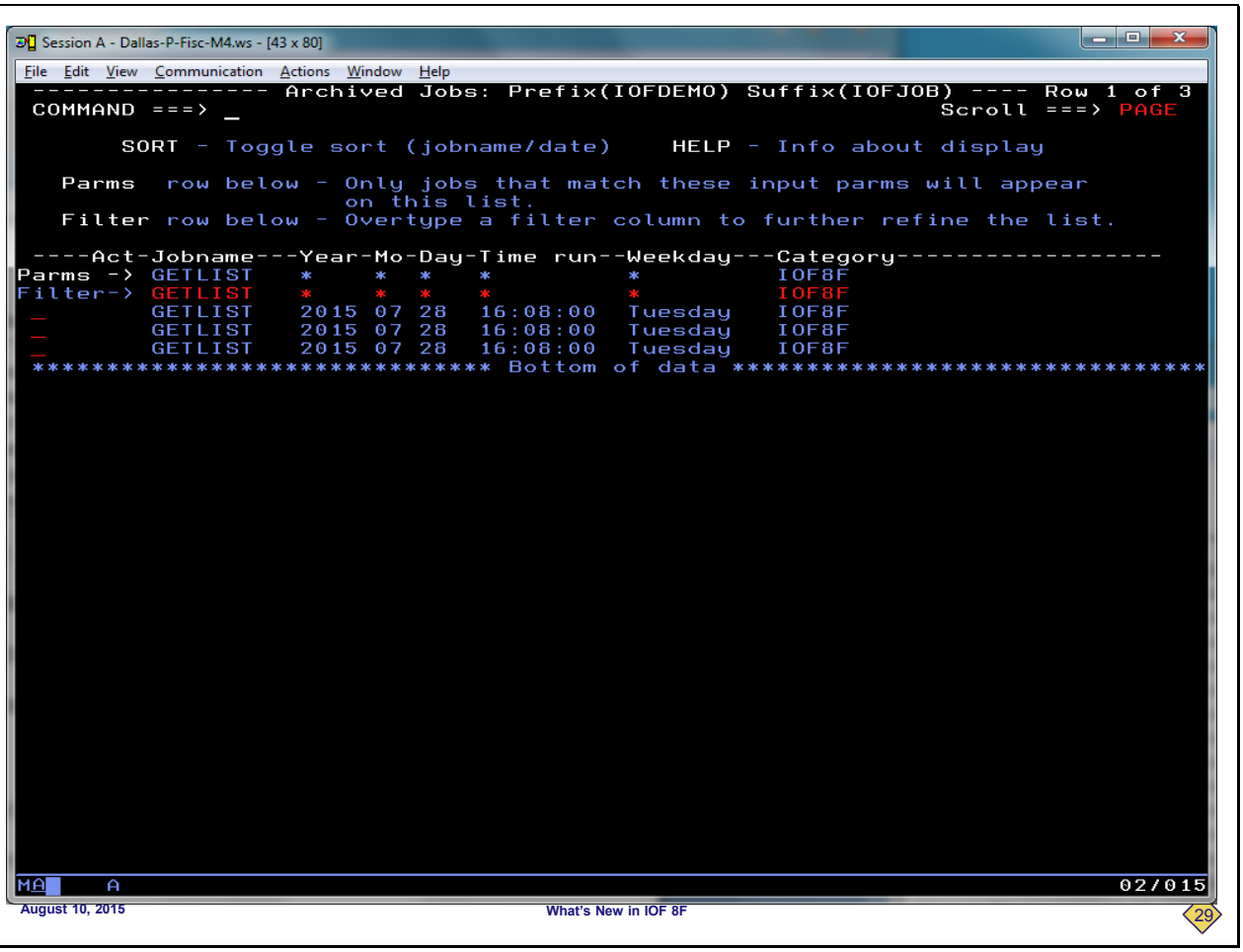
This is the interface panel for reviewing archived jobs. It has remembered our prefix and suffix from previous usage.
We want to find all of the compilations that were done for the source member GETLIST during the development for IOF 8E. So, we will enter "getlist" in the job name field and "iof8e" in the category field.



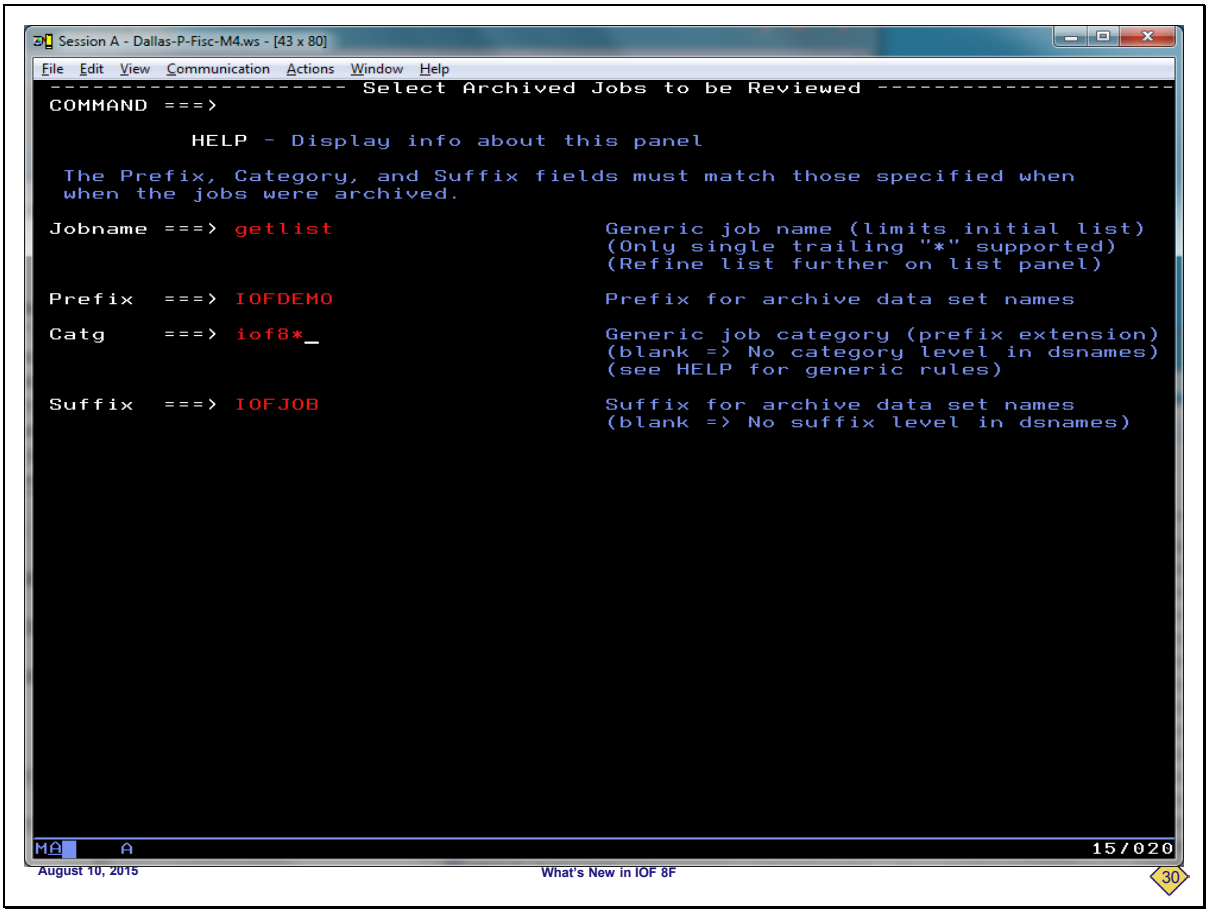
Here is a list of the compile jobs for the GETLIST source member for IOF 8E. We could select and review any of those jobs, but for the purposes of this demo, we will just return to the interface menu.



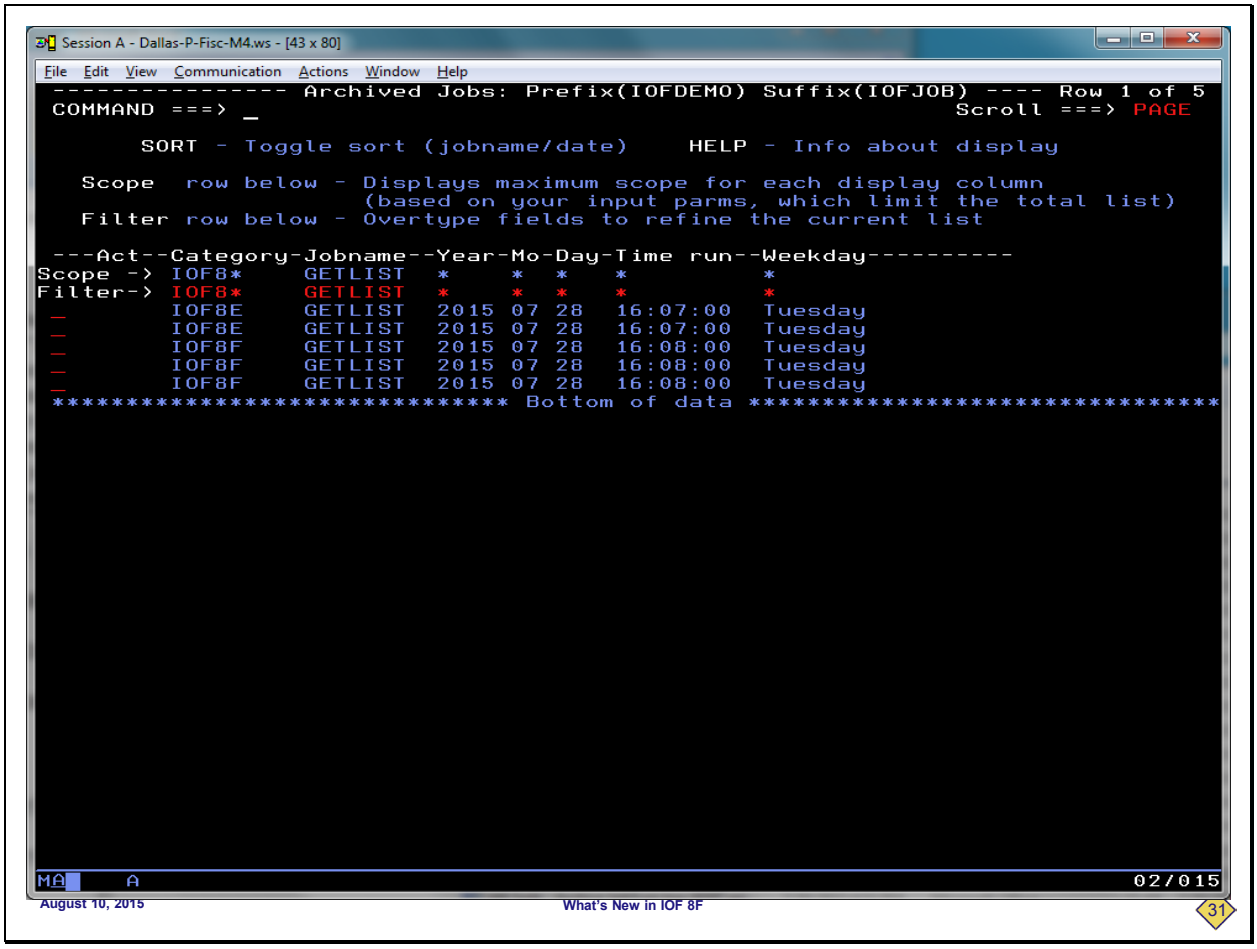
Now, we want to see the GETLIST compile jobs for IOF release 8F. We will enter "getlist" in the job name field and "iof8f" in the category field.



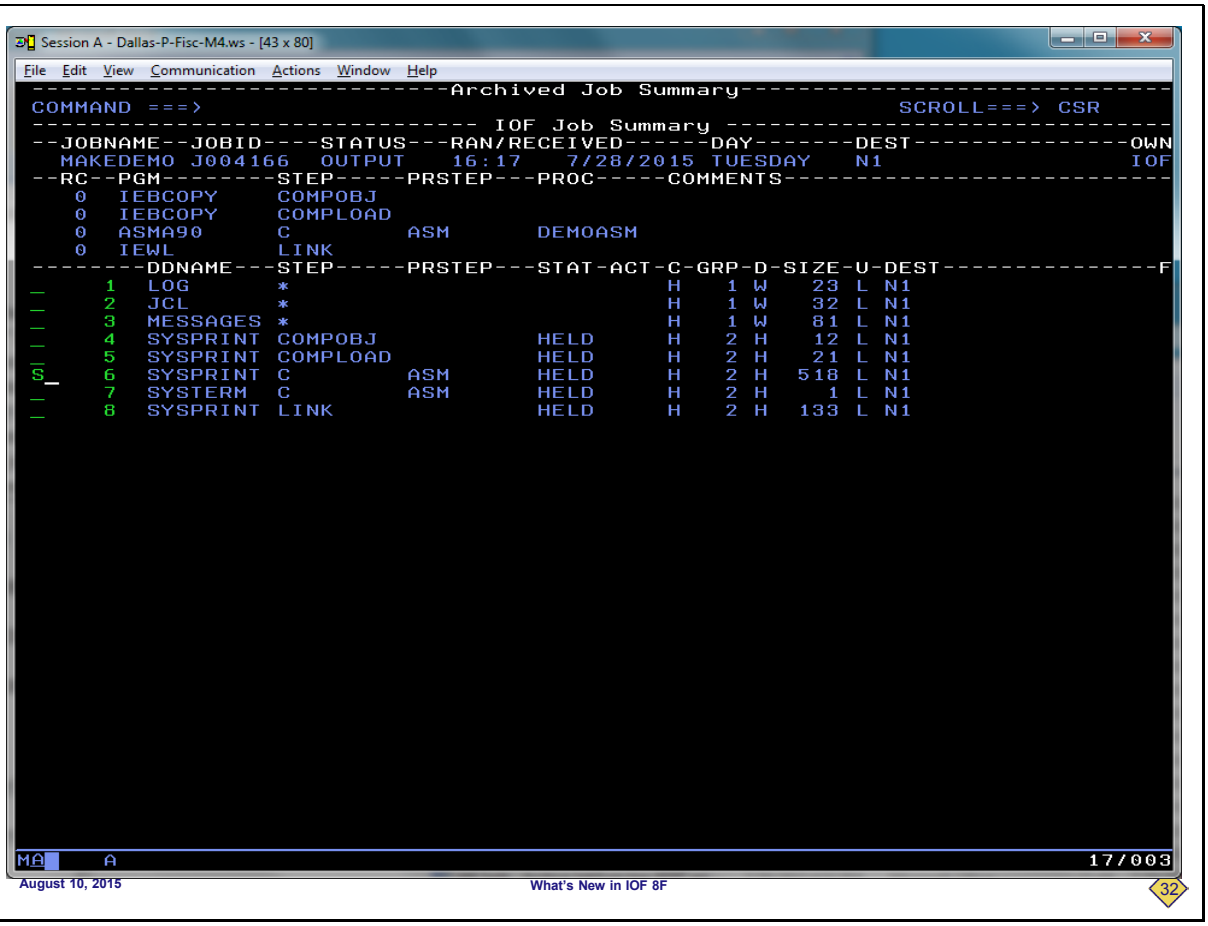
Now, we see all of the GETLIST compiles for IOF Release 8F.
We will return to the interface panel.



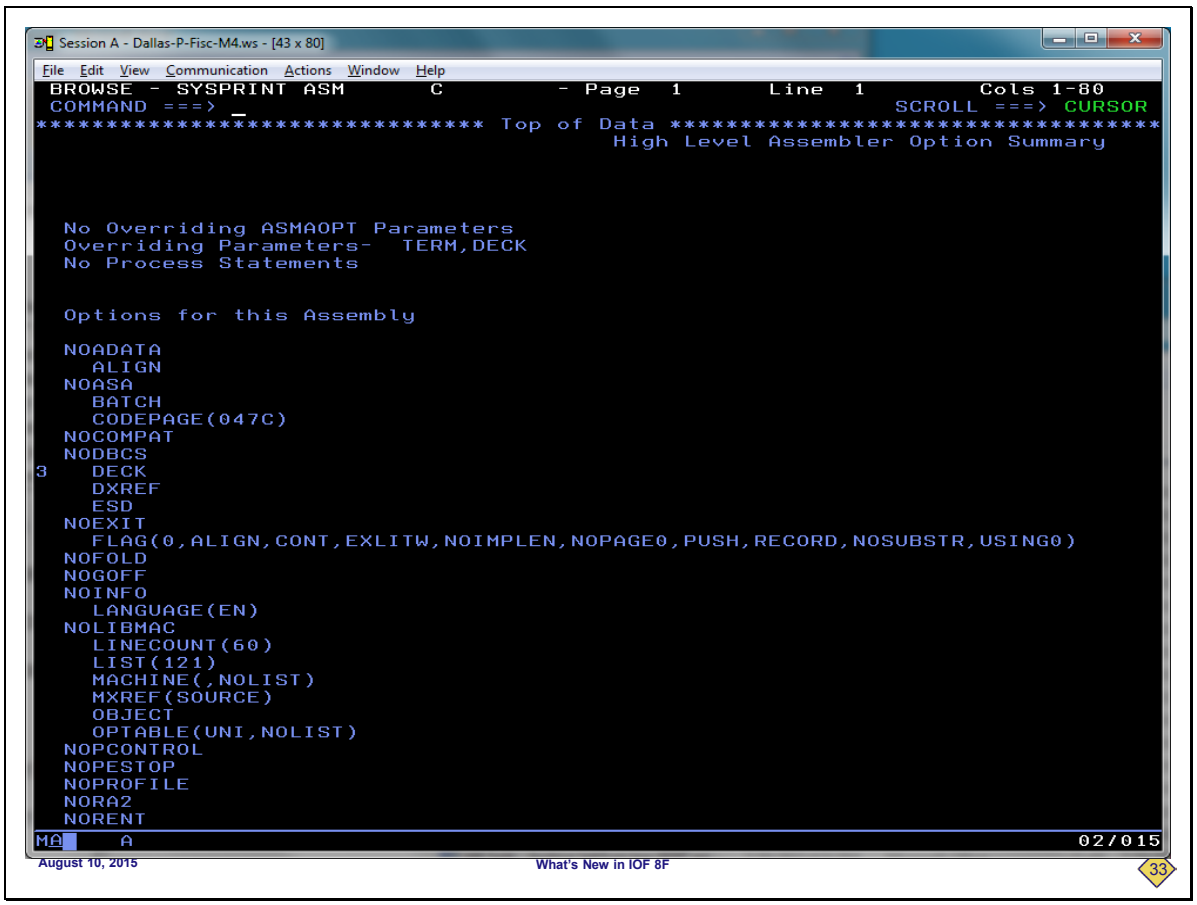
To see all of the jobs for both 8E and 8F, we will enter "getlist" in the job name field and "iof8*" in the category field.



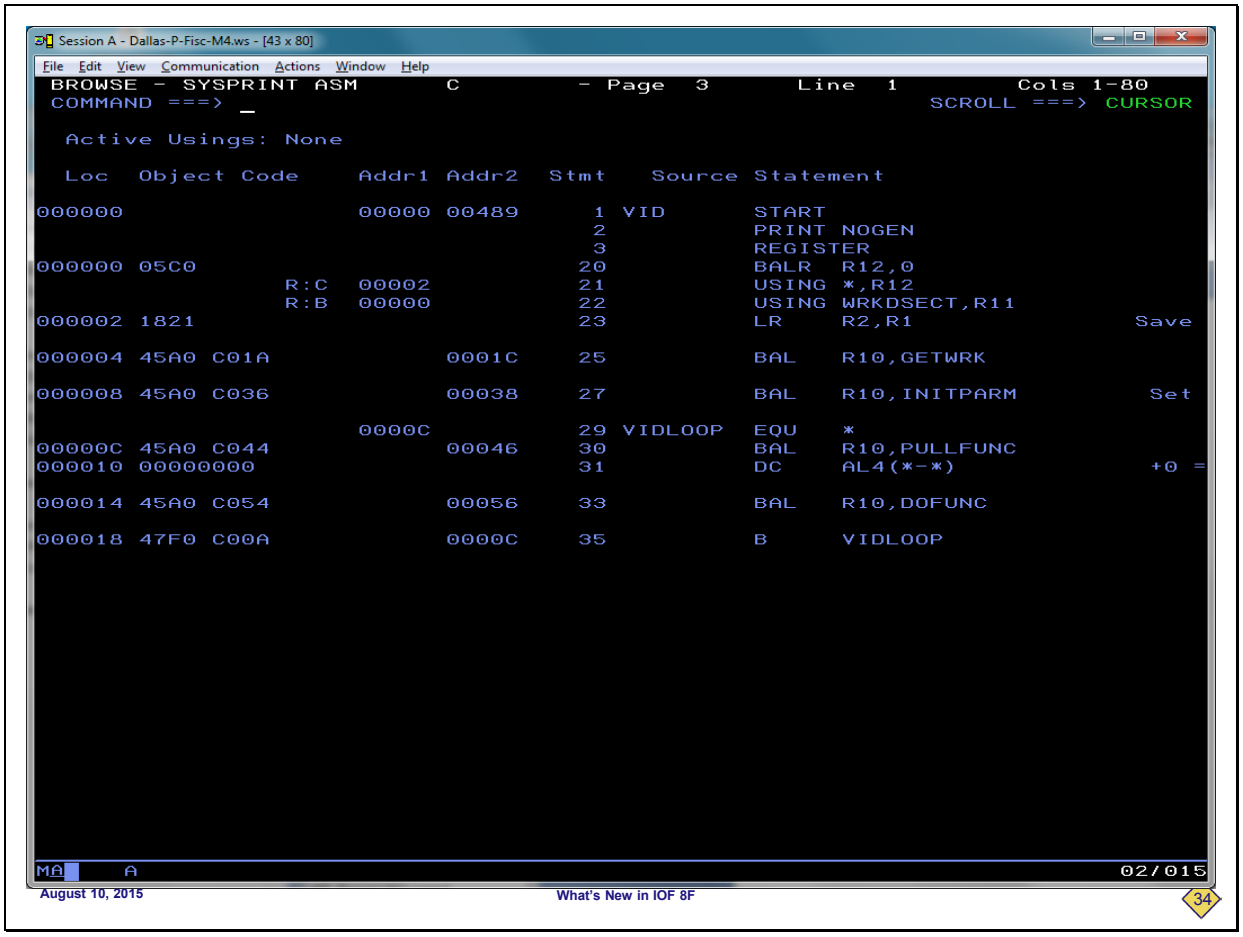
Here we can see all of the GETLIST jobs for both IOF 8E and IOF 8F. A similar application would be to archive all of your SMPE jobs for z/OS 2.1 with a category of SMPE21 and your 2.2 jobs with a category of SMPE22. Then, you could easily go back at any time in the future and find those jobs.
I hope we have shown you how simple it is to automatically archive your jobs and how useful it can be to organize them into categories.



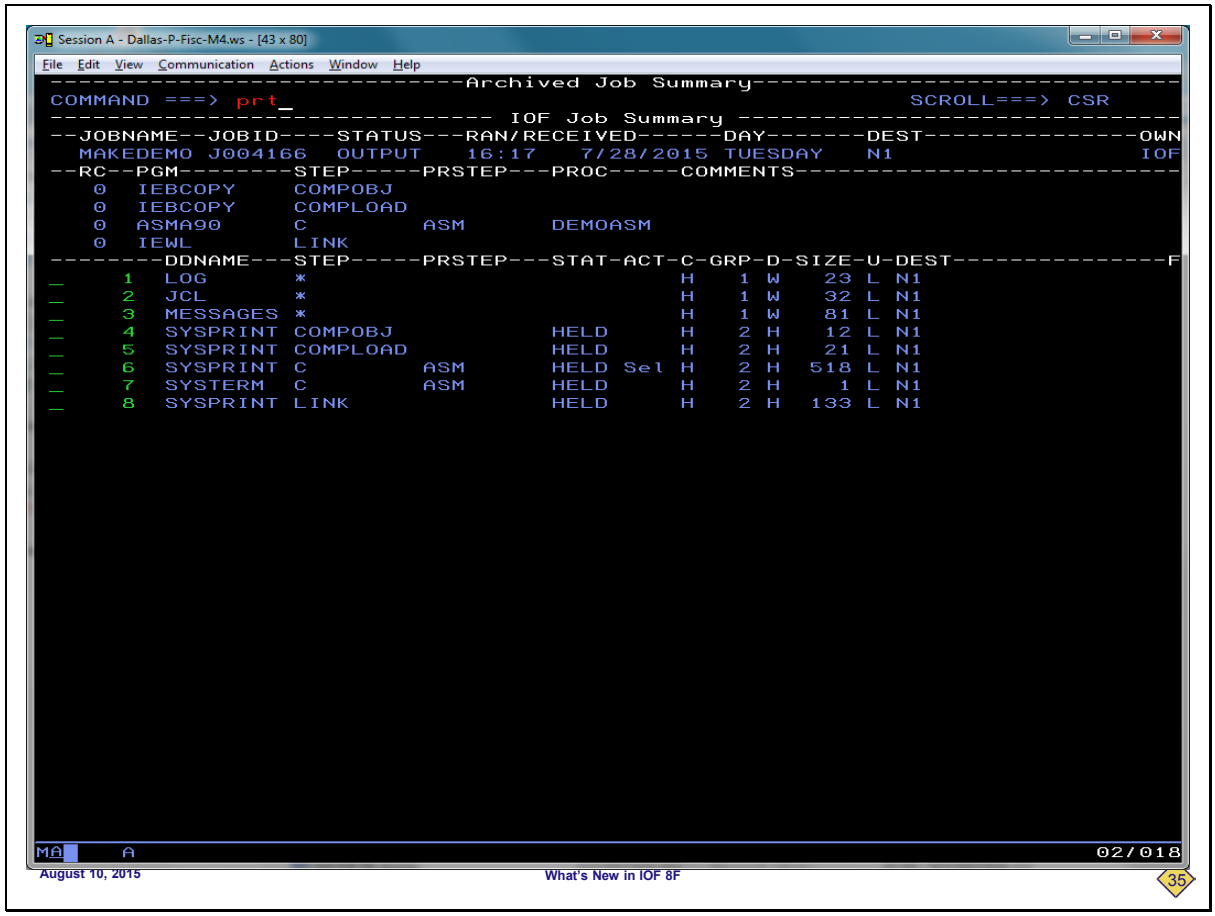
We have returned to an *Archive Job Summary* to discuss some of the IOF/JAMS enhancements for IOF 8F. All of these changes were driven by user requirements. One of the first requirements was to browse the archived sysout data sets with IOF browse instead of ISPF browse or view. Users wanted the same carriage control visual fidelity and other features of IOF browse.
To demonstrate this enhancement we will select a sysout data set from this *Archive Job Summary*.



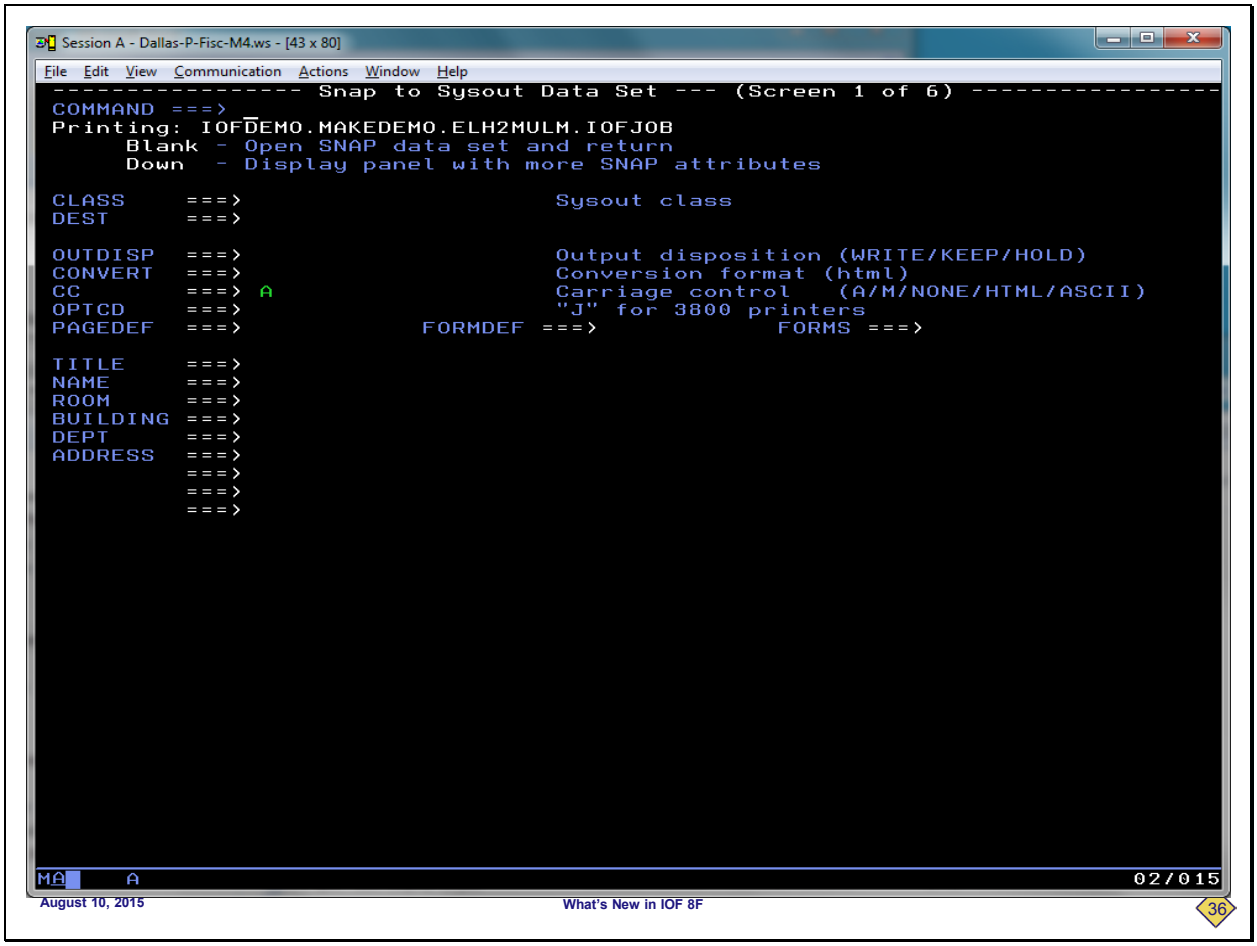
You can see that this looks a lot like IOF browse. But, that is even more obvious if we scroll down.



On this slide you can clearly see that IOF browse is honoring the carriage control in the archived sysout data set to display the data just as it would appear on a printer.
Now, we will return to the *Archive Job Summary*.

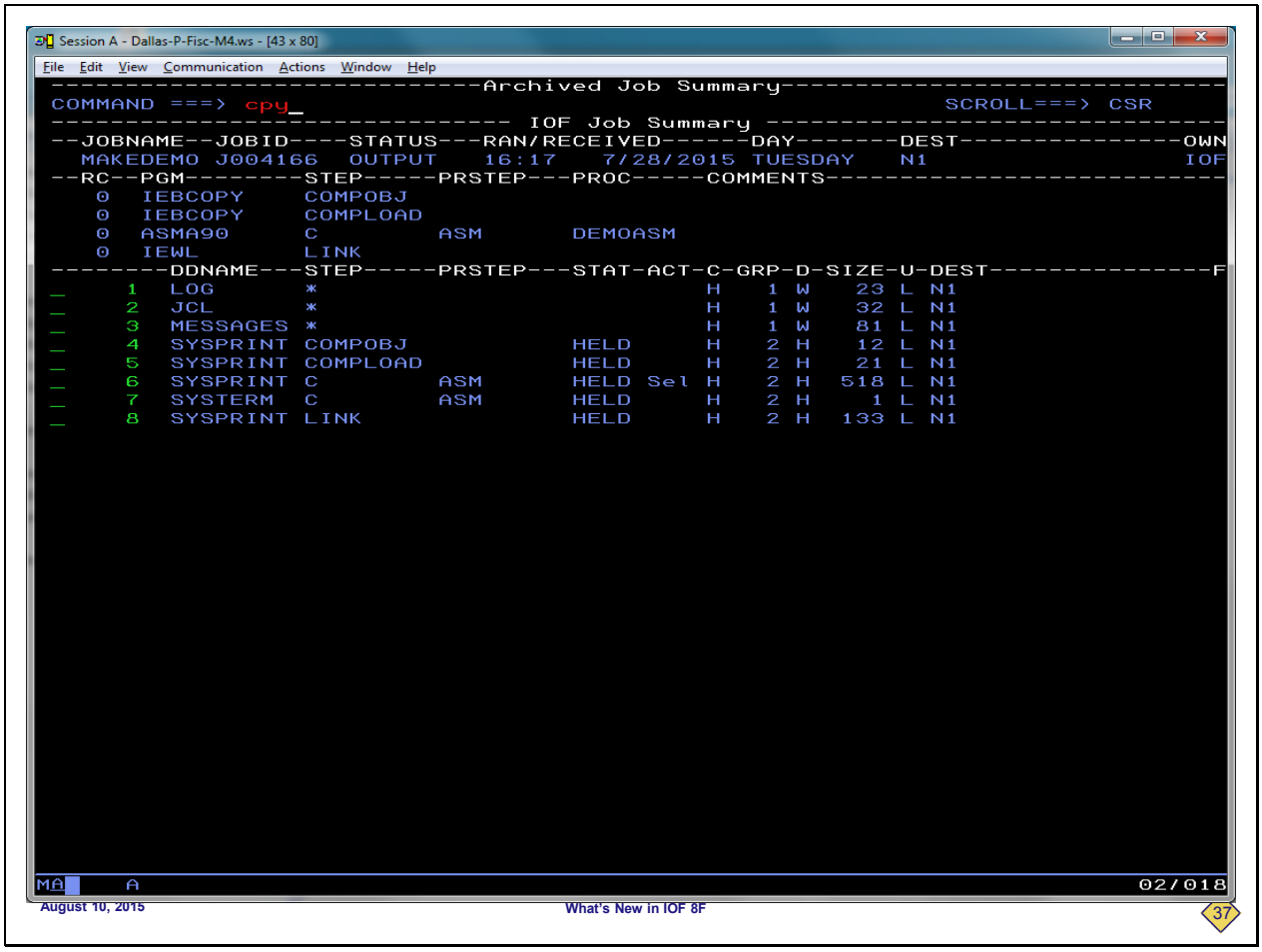


The original IOF/JAMS allowed you to print individual sysout data sets from the *Archived Job Summary,* but users wanted the ability to print the entire job.
To demonstrate that feature we will enter the PRT command

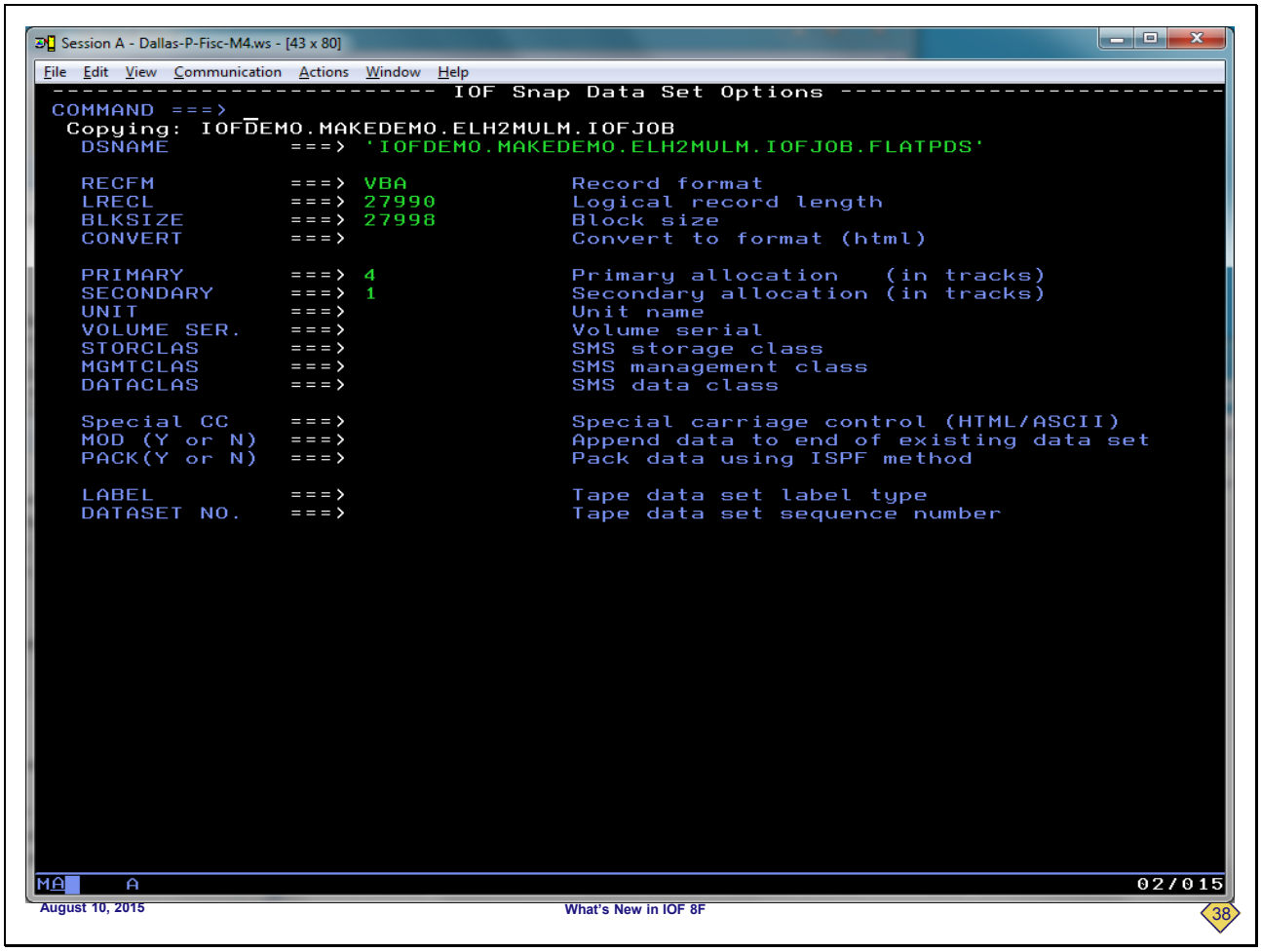


This is a standard IOF SS prompt panel, which means that you can specify virtually any JES2 print characteristics. If we press ENTER on the panel, the entire archived job would be printed using the SS characteristics.

But, we will just return to the Archive Job Summary.

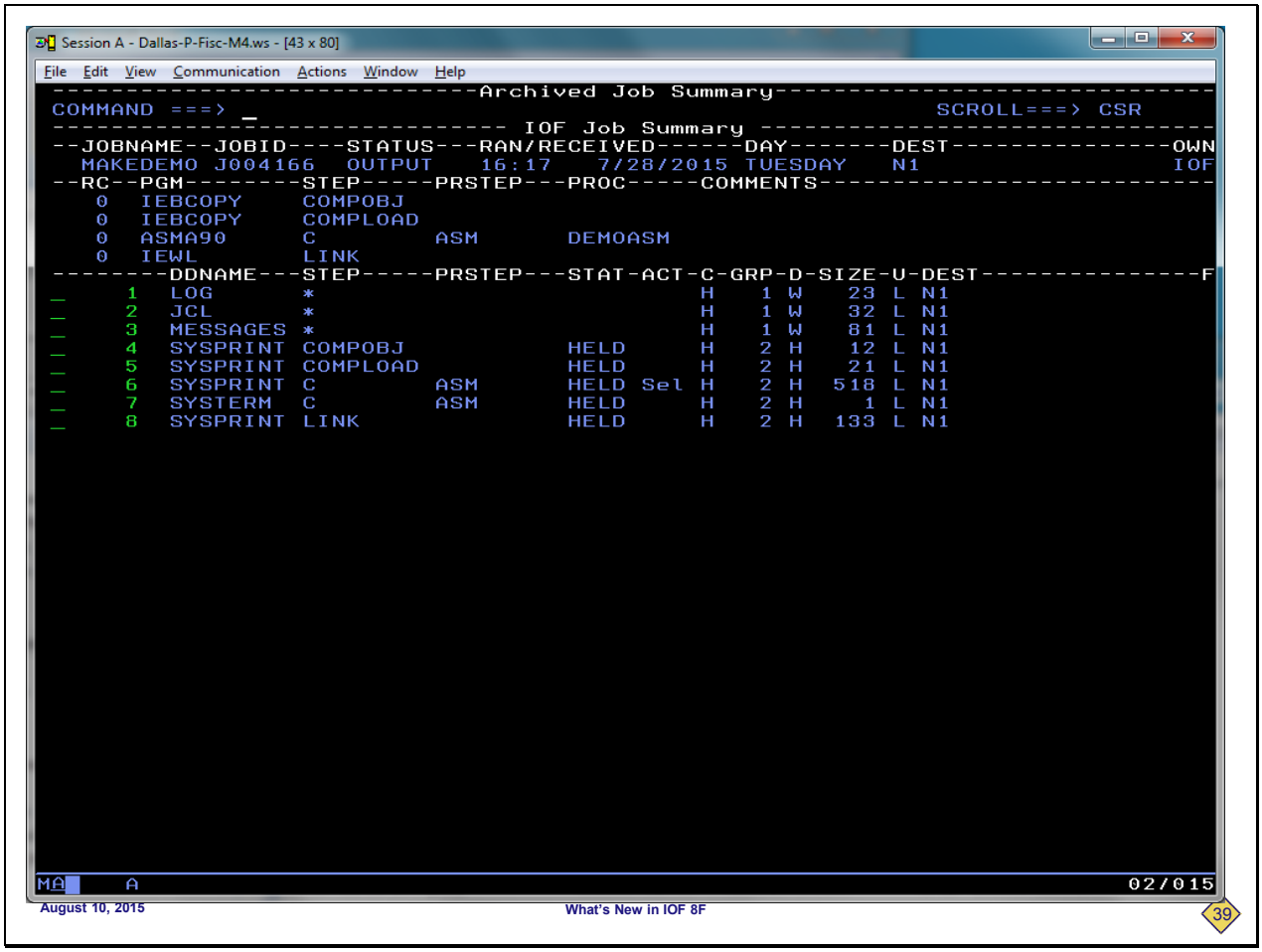


Users also said they had the need to ship off jobs to other vendors, so they needed the ability to copy the entire archived job into a sequential data set.
To demonstrate that feature we will enter the CPY command.



This is a standard IOF SD prompt panel for writing data to a sequential data set. There is a suggested data set name, but you can overtype that name. If we press ENTER on the panel, the entire archived job would be copied to the sequential data set specified on the SD panel.

But, we will just return to the *Archive Job Summary*.



Users also wanted the ability to email the entire job directly. The new SND command provides that support, but we will not demonstrate it here.
From what we have covered so far, it should be clear that IOF now has the ability to browse z/OS data sets. The archived sysouts are just members of an archive data set. Due to the very general design of the IOF browse interface, we have always known that it would not be difficult to use it to browse z/OS data sets. But before IOF/JAMS, there was never a major motivation to do that.
To implement the browsing of archived data sets, IOF 8F has an entirely new function designed to deal with z/OS data sets. It started out just browsing data sets, but other major functions were so easy to do that it expanded into a very general utility. The new ZDS command is described in a separate document.

So, IOF/JAMS is a very powerful archival system that is easy to use without help from tech support. And there are many new enhancements for 8F.