Understanding Batch Processing

Objectives

Understanding Batch Processing

Managing data is what data centers are all about, and there are many different ways in which you may need to interact with data. In this module you will see how mainframe data is stored, and how batch processing can be used to access it.

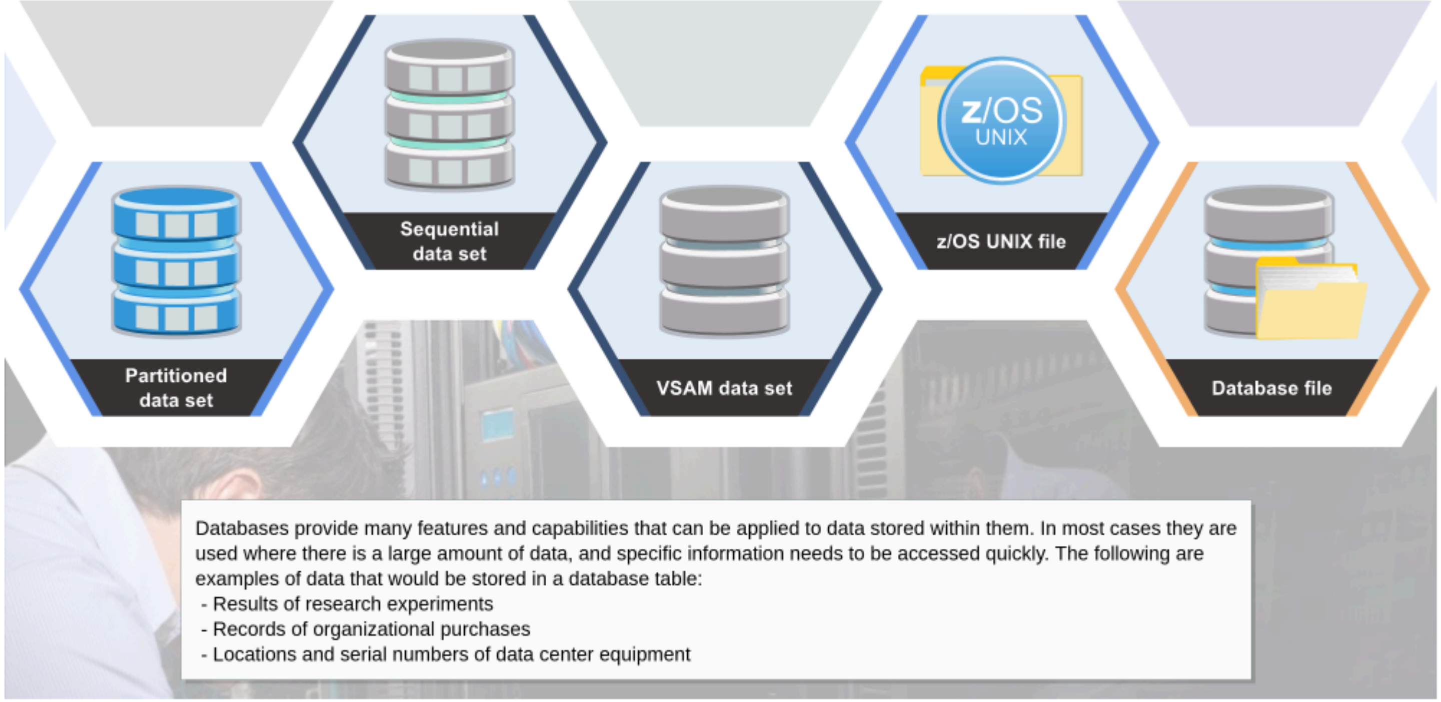
At the end of this module, you should be able to:

- Describe the Differences Between Online and Batch Processing
- Identify the z/OS Components Used in Creating and Processing a Batch Job

In today's world of business, technology has become so advanced it is now able to collect, analyze, and act upon huge amounts of data that provide them with insights into customers' needs. For this to happen, data needs to be stored in files, or data sets, and your organization's programmers need to write code that will access that data, and manipulate it in some way to produce meaningful information.

In this course, you will see how batch processing, and in particular job control language (JCL) is used to access existing data, invoke programs, and produce new data, or other types of output.

Mouse-over the job roles to see how these people may use JCL within your organization.



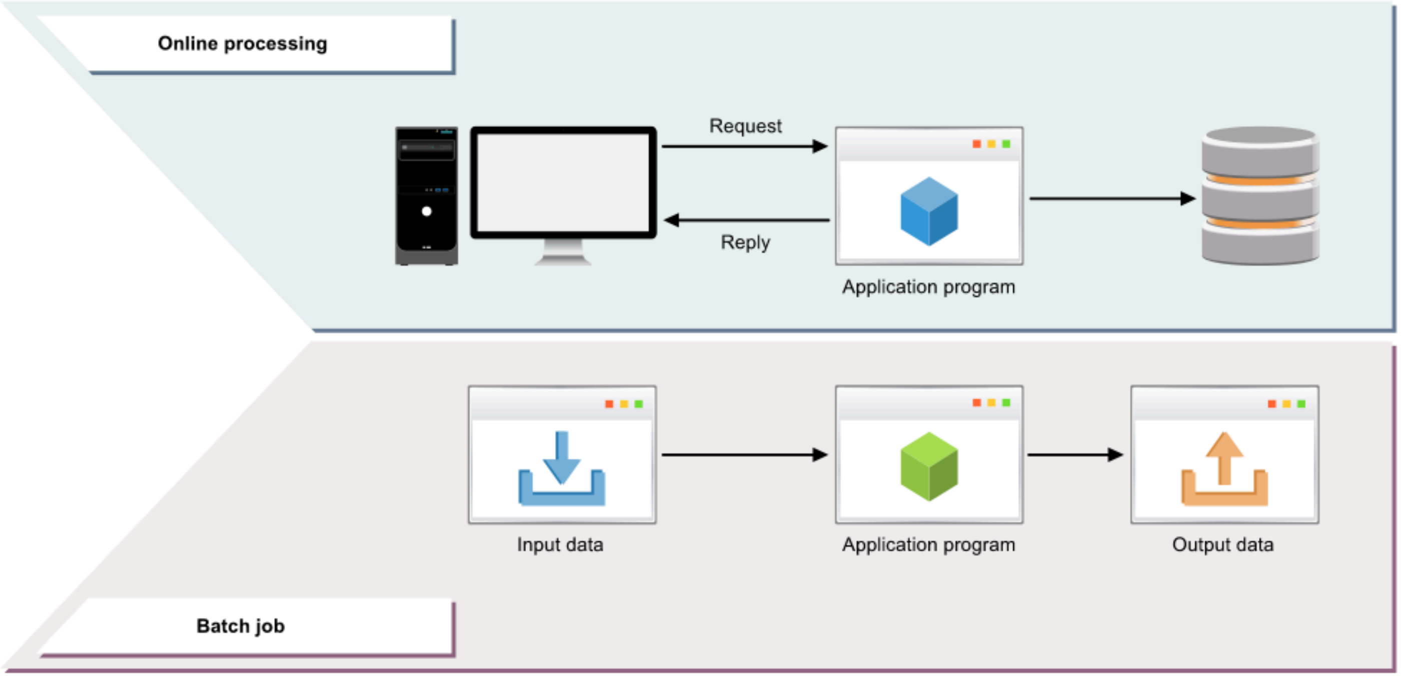
Databases provide many features and capabilities that can be applied to data stored within them. In most cases they are used where there is a large amount of data, and specific information needs to be accessed quickly. The following are examples of data that would be stored in a database table:

- Results of research experiments
- Records of organizational purchases
- Locations and serial numbers of data center equipment

What you will notice about data on the mainframe is that it does not come in one shape. For example, every record within a data set containing daily transactions will need to be processed, while other data sets provide the ability to update individual records only.

As data can be created with many different attributes, it needs to be stored in the most efficient way. You may have already encountered some of the types of data shown here, and you will see throughout these courses how JCL is used to create and reference each.

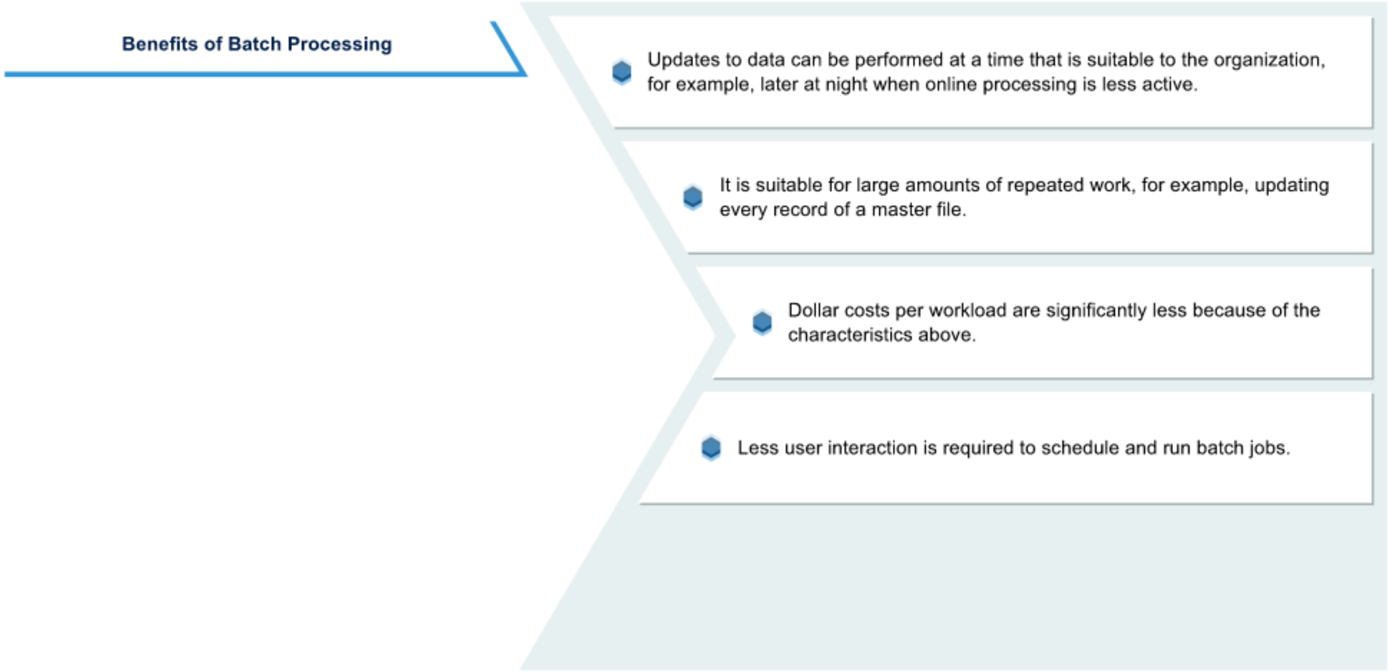
Mouse-over each data set to see how it is used.



With so many types of data on the mainframe, and many people needing to access it, how does the z/OS system manage this?

Basically, there are two methods for how this can be achieved: online and batch. Many of you will be familiar with online processing - enter a request from a screen and press Enter, the relevant data is accessed, and a response is returned to your screen. Online processing is usually reserved for simple, quick tasks.

Batch processing is often used when large amounts of data need to be accessed and worked on, usually at a pre-determined time, unlike online processing which is immediate. Batch processing uses JCL to identify the programs to be run, what is to be used as input data, and what needs to be produced as a result.



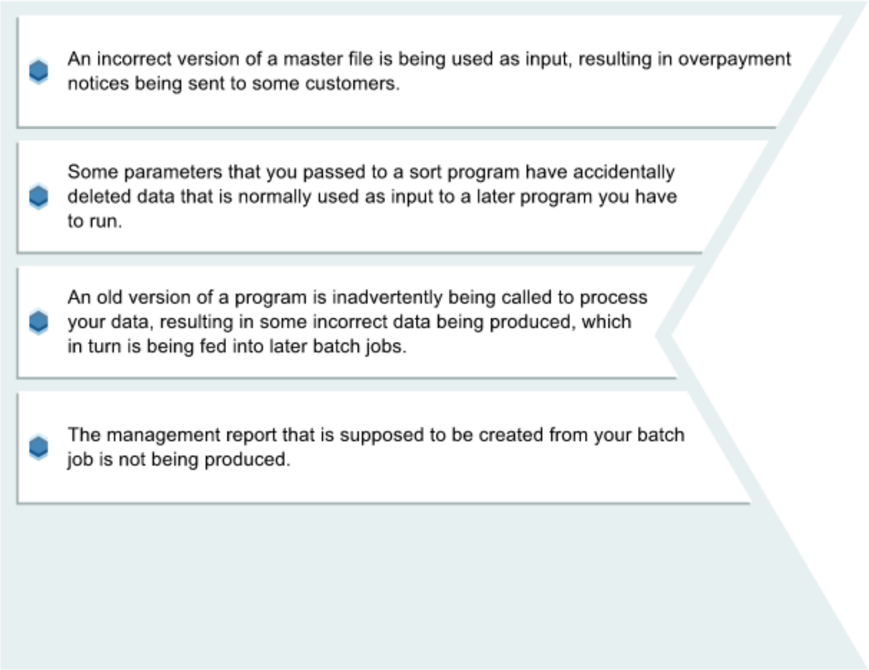
Benefits of Batch Processing

- Updates to data can be performed at a time that is suitable to the organization, for example, later at night when online processing is less active.
- It is suitable for large amounts of repeated work, for example, updating every record of a master file.
- Dollar costs per workload are significantly less because of the characteristics above.
- Less user interaction is required to schedule and run batch jobs.

Why would you want to use batch processing, if online processing can perform tasks instantly?

Even though you live in a world where customers expect instant results, the reality is that the cost to make resources available to achieve this is not economical, at least not yet.

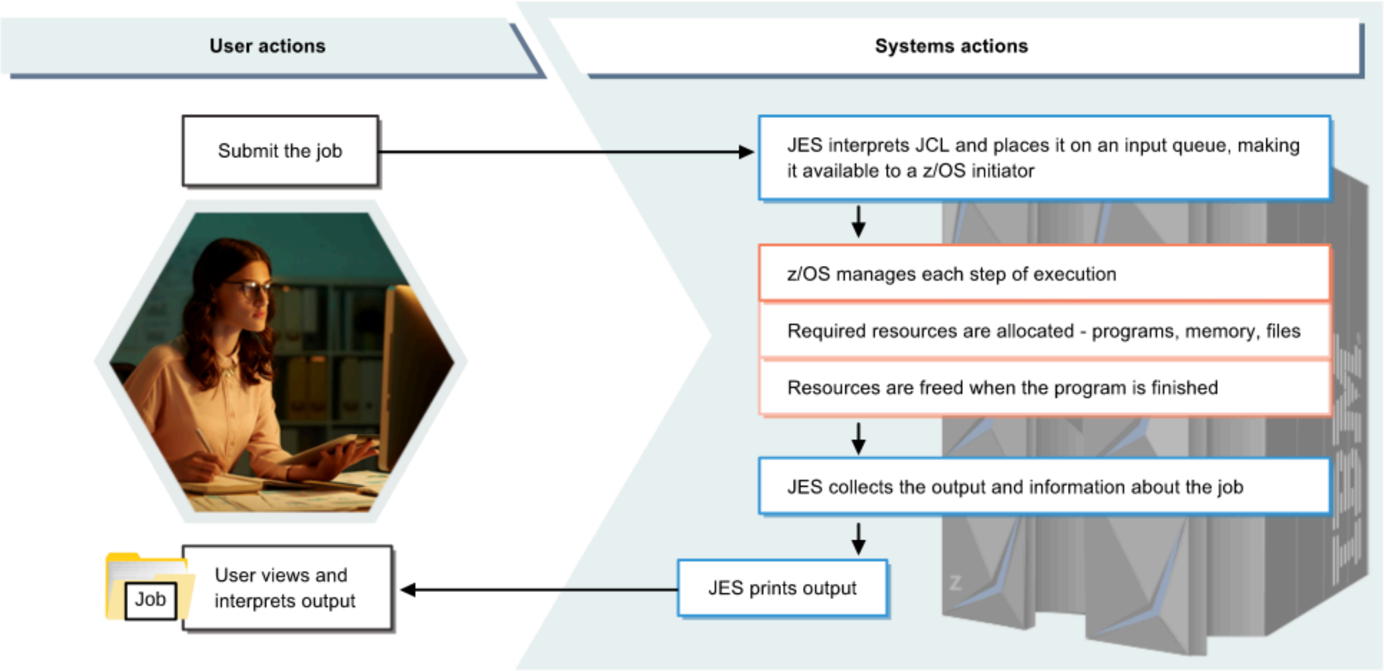
You can see here that batch processing is perfect for situations where there is repetitious, high volume work.



- An incorrect version of a master file is being used as input, resulting in overpayment notices being sent to some customers.
- Some parameters that you passed to a sort program have accidentally deleted data that is normally used as input to a later program you have to run.
- An old version of a program is inadvertently being called to process your data, resulting in some incorrect data being produced, which in turn is being fed into later batch jobs.
- The management report that is supposed to be created from your batch job is not being produced.

The batch processing discussed on previous pages focuses primarily with the production responsibilities of the business, for example, a job that runs every day or night that applies payment related transactions to customer data. The JCL used for this purpose will have been rigorously tested before being accepted into production, to avoid errors and problems.

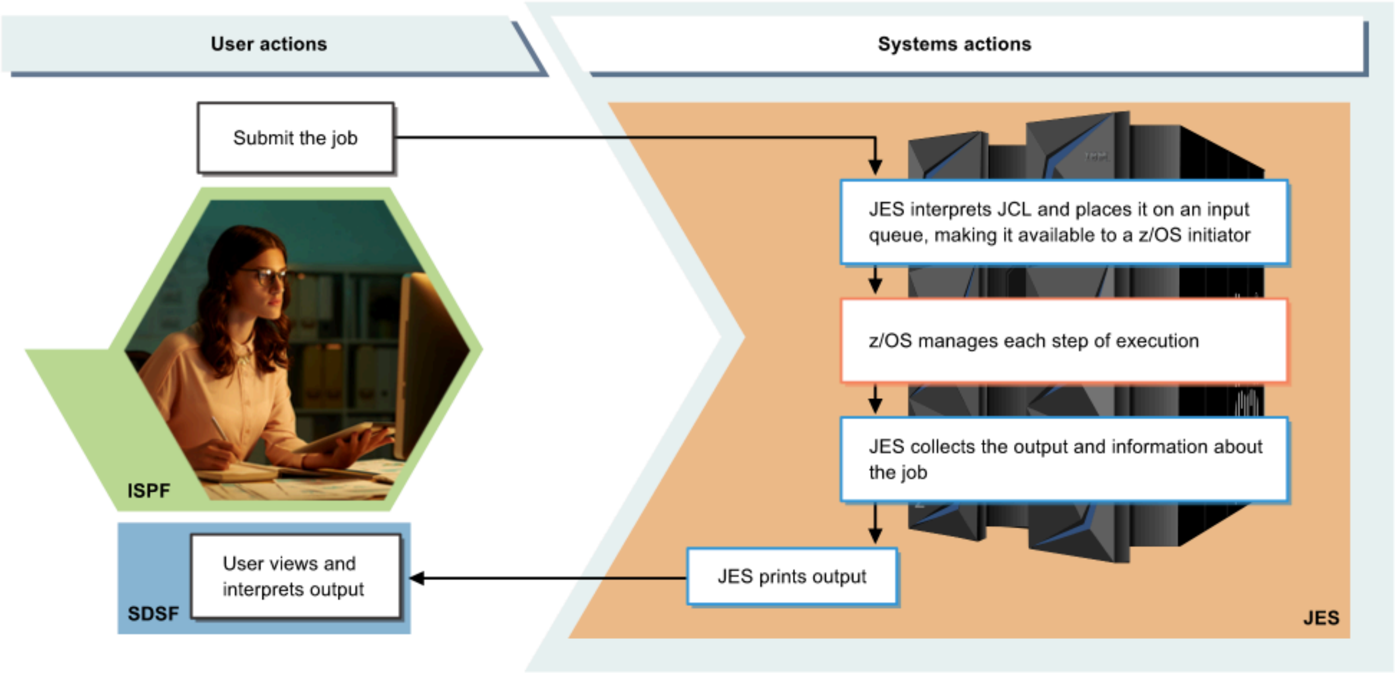
Your relationship with batch processing is likely to be different. In your situation you might need to run a batch job often to compile a COBOL program, run an ad hoc job to produce management reports, or to test programs before they move into the production environment. It is not likely to be a time-consuming task working with JCL, however if it is not coded correctly, it can have some drastic consequences.



User actions
Systems actions
Submit the job
JES interprets JCL and places it on an input queue, making it available to a z/OS initiator
z/OS manages each step of execution
Required resources are allocated - programs, memory, files
Resources are freed when the program is finished
JES collects the output and information about the job
JES prints output
Job
User views and interprets output

When your batch job is submitted to the system, what actually happens to it? Being aware of this process can help you at a later stage to diagnose JCL error messages.

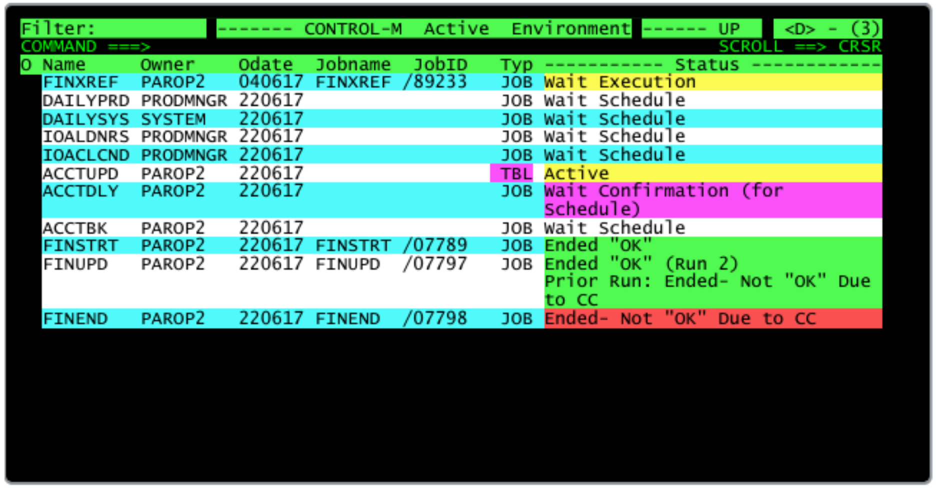
Click **Play** to see what happens when a batch job is submitted for processing.


User actions
Systems actions
Submit the job
JES interprets JCL and places it on an input queue, making it available to a z/OS initiator
z/OS manages each step of execution
JES collects the output and information about the job
JES prints output
User views and interprets output
ISPF
SDSF
JES

Now that you can see the underlying processes behind the running of a batch job, the following are types of skills and knowledge, apart from JCL, that you need to create and manage your own batch job:

- ISPF - to access the JCL code that is used for a batch job
- JES - you will need some background on how JES, JES2, or JES3 is going to handle your submitted batch job
- SDSF or similar output viewing software - to display the results following the completion of your batch job

If you are unfamiliar with these products, you should revisit what Interskill offers in these areas.

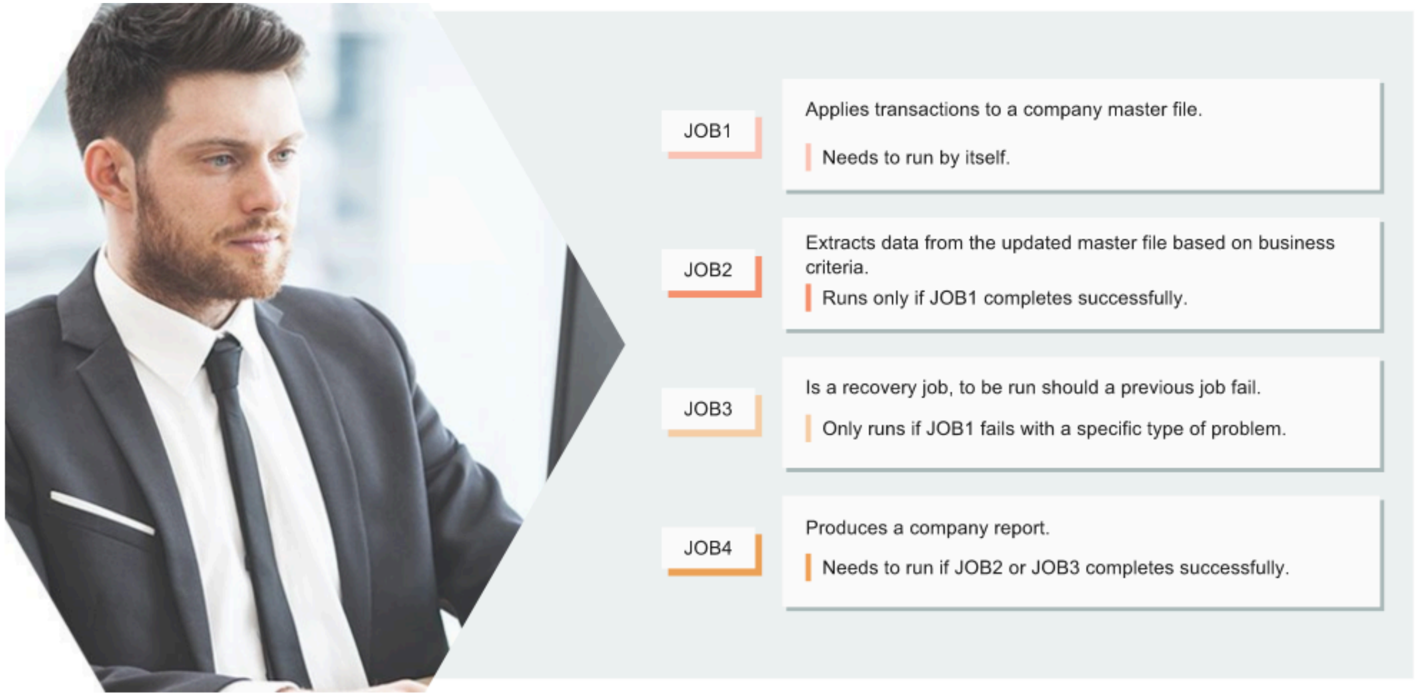


```
Filter:              ------- CONTROL-M  Active  Environment ------ UP   <D> - (3)
COMMAND ===>                                                     SCROLL ===> CRSR
O Name     Owner    Odate  Jobname  JobID    Typ ----------- Status -----------
  FINXREF  PAROP2   040617 FINXREF /89233   JOB Wait Execution
  DAILYPRD PRODMNGR 220617                  JOB Wait Schedule
  DAILYSYS SYSTEM   220617                  JOB Wait Schedule
  IOALDNRS PRODMNGR 220617                  JOB Wait Schedule
  IOACLCND PRODMNGR 220617                  JOB Wait Schedule
  ACCTUPD  PAROP2   220617                  TBL Active
  ACCTDLY  PAROP2   220617                  JOB Wait Confirmation (for
                                                Schedule)
  ACCTBK   PAROP2   220617                  JOB Wait Schedule
  FINSTRT  PAROP2   220617 FINSTRT /07789   JOB Ended "OK"
  FINUPD   PAROP2   220617 FINUPD  /07797   JOB Ended "OK" (Run 2)
                                                Prior Run: Ended- Not "OK" Due
                                                to CC
  FINEND   PAROP2   220617 FINEND  /07798   JOB Ended- Not "OK" Due to CC
```

This capture is from the BMC Control-M product, which is one of several enterprise scheduling products used by large organizations. Other similar products include IBM Workload Scheduler and CA Workload Automation - CA 7 Edition.

Many larger organizations have implemented automated batch scheduling software that is capable of handling complex batch job scheduling requirements. For example, batch processing may need to run at a specific time, once other data has become available, when a resource becomes free, or simply triggered once another batch job has completed.

In this environment, you may be required to work with this software to schedule your own jobs. In other situations, there may be one or more production schedulers whose role is to define the scheduling of all batch work across the business, using this product.



JOB1

Applies transactions to a company master file.

Needs to run by itself.

JOB2

Extracts data from the updated master file based on business criteria.

Runs only if JOB1 completes successfully.

JOB3

Is a recovery job, to be run should a previous job fail.

Only runs if JOB1 fails with a specific type of problem.

JOB4

Produces a company report.

Needs to run if JOB2 or JOB3 completes successfully.

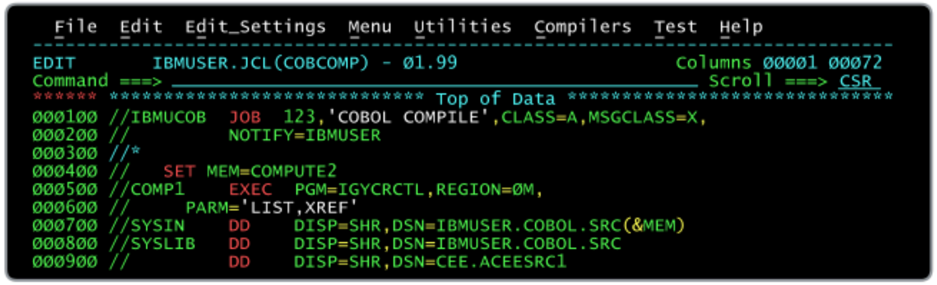
As of z/OS 2.2, you can create simple scheduling for your batch jobs, within the batch job itself. This JCL code is referenced by JES, which can run jobs at specific times, or after other jobs have completed.

Click Play to see how these jobs need to be scheduled, and how this could be achieved using JCL.

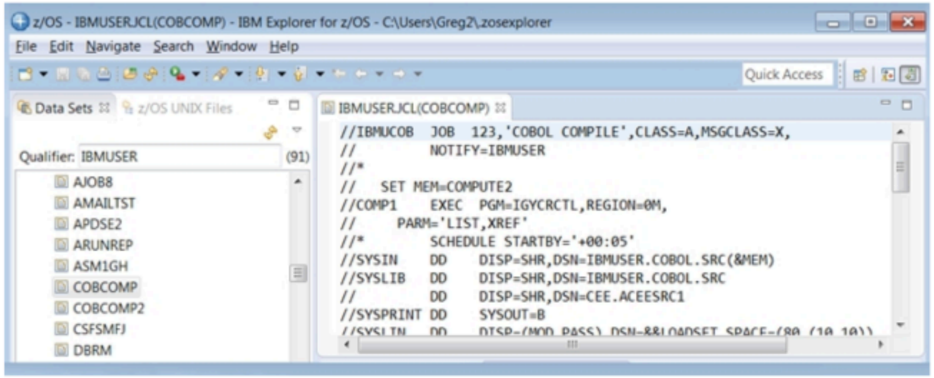
- You are aware of the people in your organization who use JCL, and for what purposes.
- Mainframe data is stored in a variety of formats.
- There are two types of data processing: online and batch.
- To work with JCL, you also need skills in ISPF and SDSF use, and knowledge of JES processing.
- Batch processing can be initiated manually, or through automated scheduling software.

In this section you have seen that many people in your organization use JCL to access all types of data, to run programs, and produce some valued output. You looked at processes involved in running a batch job, and the skills and knowledge that you require to create and manage your batch job JCL code.

In the next section you will look at some prerequisites before you start to create and run your own JCL batch job.



```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
 -------------------------------------------------------------------------------
 EDIT       IBMUSER.JCL(COBCOMP) - 01.99                   Columns 00001 00072
 Command ===> ________________________________________________ Scroll ===> CSR
 ****** ***************************** Top of Data ******************************
 000100 //IBMUCOB  JOB  123,'COBOL COMPILE',CLASS=A,MSGCLASS=X,
 000200 //            NOTIFY=IBMUSER
 000300 //*
 000400 //    SET MEM=COMPUTE2
 000500 //COMP1     EXEC  PGM=IGYCRCTL,REGION=0M,
 000600 //       PARM='LIST,XREF'
 000700 //SYSIN     DD     DISP=SHR,DSN=IBMUSER.COBOL.SRC(&MEM)
 000800 //SYSLIB    DD     DISP=SHR,DSN=IBMUSER.COBOL.SRC
 000900 //          DD     DISP=SHR,DSN=CEE.ACEESRC1
```


z/OS - IBMUSER.JCL(COBCOMP) - IBM Explorer for z/OS - C:\Users\Greg2\.zosexplorer
File Edit Navigate Search Window Help
Quick Access
Data Sets
z/OS UNIX Files
IBMUSER.JCL(COBCOMP)

Qualifier: IBMUSER (91)

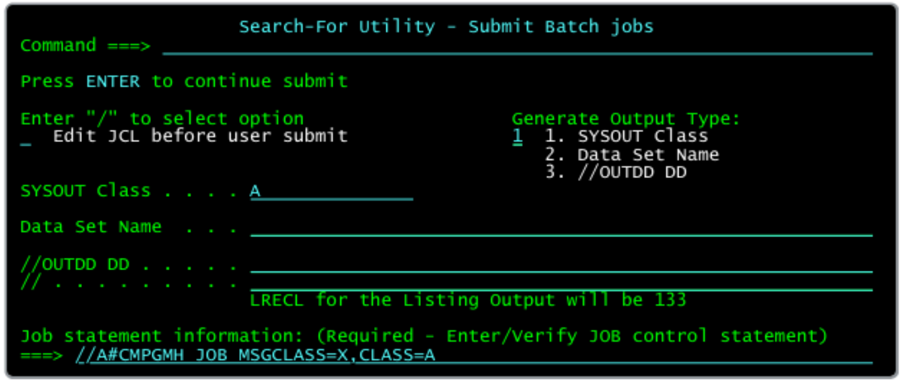
- AJOB8
- AMAILTST
- APDSE2
- ARUNREP
- ASM1GH
- COBCOMP
- COBCOMP2
- CSFSMFJ
- DBRM

```
//IBMUCOB  JOB  123,'COBOL COMPILE',CLASS=A,MSGCLASS=X,
//         NOTIFY=IBMUSER
//*
//   SET MEM=COMPUTE2
//COMP1    EXEC  PGM=IGYCRCTL,REGION=0M,
//     PARM='LIST,XREF'
//*        SCHEDULE STARTBY='+00:05'
//SYSIN    DD    DISP=SHR,DSN=IBMUSER.COBOL.SRC(&MEM)
//SYSLIB   DD    DISP=SHR,DSN=IBMUSER.COBOL.SRC
//         DD    DISP=SHR,DSN=CEE.ACEESRC1
//SYSPRINT DD    SYSOUT=B
//SYSLIN   DD    DISP=(MOD,PASS),DSN=&&LOADSET,SPACE=(80,(10,10))
```

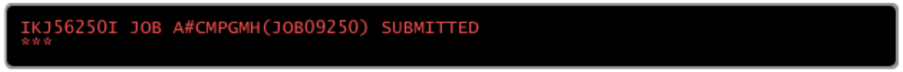
You now know that batch processing can be used to access mainframe data, but where do you start?

The first step is to become familiar with the interface used for submitting batch jobs to the system. While the ISPF interface shown here will be familiar to many of you, more intuitive interfaces such as the IBM Explorer for z/OS GUI provides syntax highlighting and checking, as well as auto completion facilities.

The majority of this course focuses on using JCL from the traditional screens.



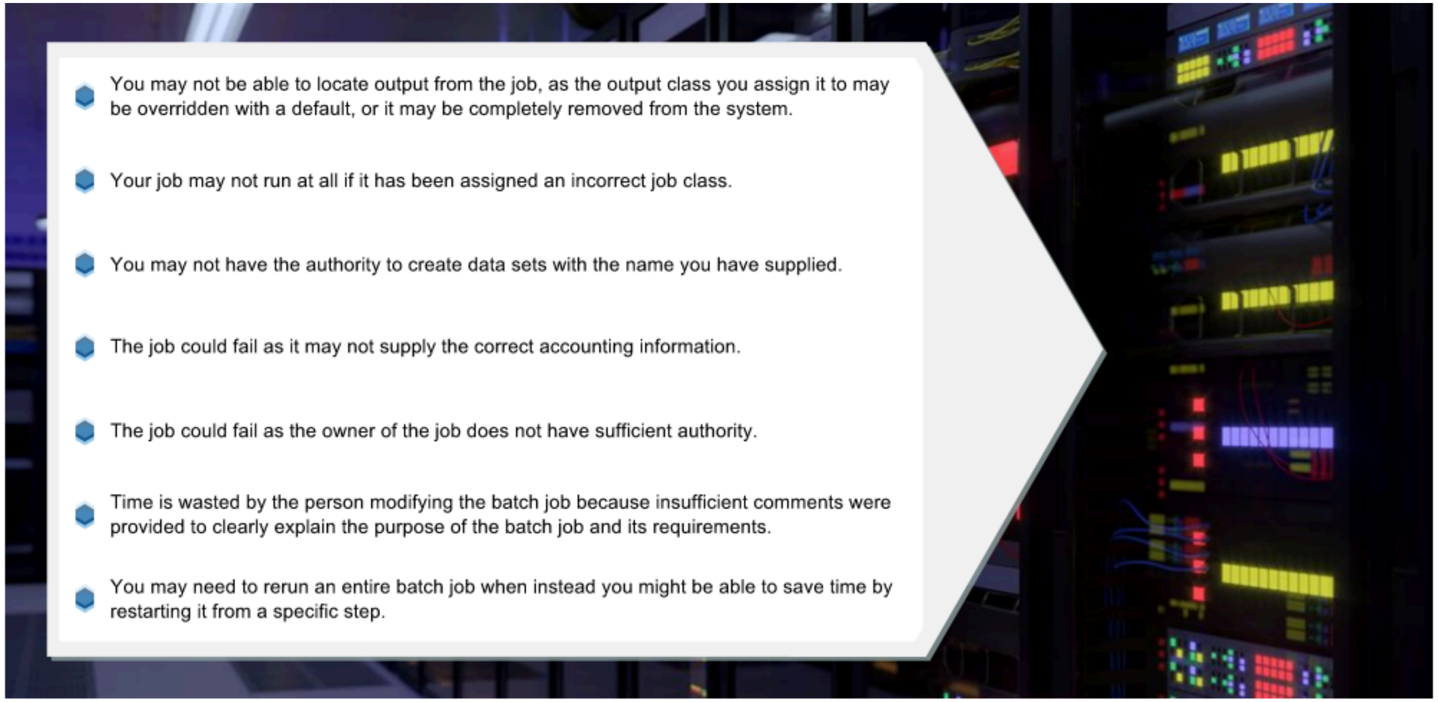
```
                    Search-For Utility - Submit Batch jobs
Command ===>

Press ENTER to continue submit

Enter "/" to select option                   Generate Output Type:
_  Edit JCL before user submit               1  1. SYSOUT Class
                                                2. Data Set Name
                                                3. //OUTDD DD

SYSOUT Class . . . . A

Data Set Name  . . .

//OUTDD DD . . . . .
// . . . . . . . . .
                     LRECL for the Listing Output will be 133

Job statement information: (Required - Enter/Verify JOB control statement)
===> //A#CMPGMH JOB MSGCLASS=X,CLASS=A
```



```
IKJ56250I JOB A#CMPGMH(JOB09250) SUBMITTED
***
```

Your exposure to JCL may be varied. For example, you may work in an area where someone has created a utility where you just need to add your input and output requirements, the program you need to run, and you press a button that will create and submit your batch job. There are also several ISPF options such as the Search-For utility shown here that will generate JCL based on the input you provide.

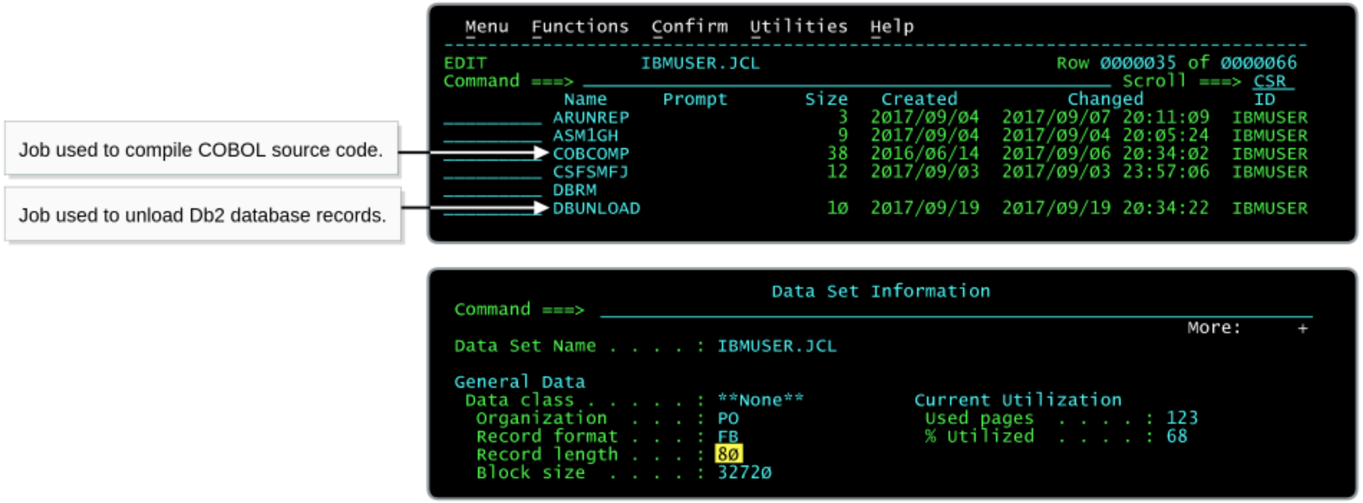
If you have the need to manually run several batch jobs or one that you intend to run regularly, you will need to ensure that it resides in a data set that you have authority to access.



- You may not be able to locate output from the job, as the output class you assign it to may be overridden with a default, or it may be completely removed from the system.
- Your job may not run at all if it has been assigned an incorrect job class.
- You may not have the authority to create data sets with the name you have supplied.
- The job could fail as it may not supply the correct accounting information.
- The job could fail as the owner of the job does not have sufficient authority.
- Time is wasted by the person modifying the batch job because insufficient comments were provided to clearly explain the purpose of the batch job and its requirements.
- You may need to rerun an entire batch job when instead you might be able to save time by restarting it from a specific step.

Before going too far, you should also check whether your organization has any batch processing standards. As well as specifying the parameters you will need to adhere to when submitting batch jobs, it may also identify the data set you need to store your JCL in.

If you do not follow your organization's batch standards, you could encounter problems such as the ones shown here.



```
 Menu  Functions  Confirm  Utilities  Help
------------------------------------------------------------------------------
EDIT              IBMUSER.JCL                            Row 0000035 of 0000066
Command ===> ________________________________________________ Scroll ===> CSR
           Name      Prompt       Size   Created          Changed          ID
_________ ARUNREP                    3  2017/09/04  2017/09/07 20:11:09  IBMUSER
_________ ASM1GH                     9  2017/09/04  2017/09/04 20:05:24  IBMUSER
_________ COBCOMP                   38  2016/06/14  2017/09/06 20:34:02  IBMUSER
_________ CSFSMFJ                   12  2017/09/03  2017/09/03 23:57:06  IBMUSER
_________ DBRM
_________ DBUNLOAD                  10  2017/09/19  2017/09/19 20:34:22  IBMUSER
```

Job used to compile COBOL source code.

Job used to unload Db2 database records.

```
                             Data Set Information
Command ===> _________________________________________________________________
                                                                 More:     +
Data Set Name . . . . : IBMUSER.JCL

General Data
 Data class . . . . . : **None**          Current Utilization
  Organization  . . . : PO                 Used pages  . . . . : 123
  Record format . . . : FB                 % Utilized  . . . . : 68
  Record length . . . : 80
  Block size  . . . . : 32720
```

You will find that the data set used for storing your batch jobs is a partitioned data set (PDS). If you remember from earlier on, a PDS contains individual members, making it an ideal location to store all batch jobs belonging to you, or your group.

When creating a new member to store your batch job, try to make its name as descriptive as you can, a maximum of eight characters are allowed for a member name, as it will make it easier at a later stage for you to locate it. If you were to look at the attributes of the PDS you would see that it has a record length of 80 characters, which is used for historical purposes.

- A traditional 3270 screen, or newer GUI interface through products such as z/OS Explorer, can be used to access JCL.
- JCL should be stored in a PDS member.
- The PDS used to store JCL must have a logical record length (LRECL) of 80.
- A descriptive name for the PDS member containing your JCL should be used.
- There are likely to be organizational standards used for JCL coding and general usage.

You now have a good understanding of where batch jobs reside and how they can be accessed. You will also be aware of organizational standards that will undoubtedly exist to enforce your use of JCL.

In the courses that follow you will begin to dissect JCL looking at the statements and parameters that can be used, and the consequences should they be coded incorrectly.

Summary

Understanding Batch Processing

Managing data is what data centers are all about, and there are many different ways in which you may need to interact with data. In this module you saw how mainframe data is stored, and how batch processing can be used to access it.

You should now be able to:

- Describe the Differences Between Online and Batch Processing
- Identify the z/OS Components Used in Creating and Processing a Batch Job

Question 2 of 7 - Enterprise Data Processing

Which product is used to create and store JCL code?

Select the correct option.

Click **Check My Answer** when you have finished.

□ ISPF

□ JES3

□ SDF

□ JES2

Check My Answer ✔