Which system is the job running on?

In June 2018 a topic was raised on LinkedIn along the lines of **"Is there some way of knowing the host environment where the JCL is running? Suppose the test environment is TESTXX and the production environment is PRODYY. I'd like to execute one of JCL's step only in TESTXX.**"

The first point to make I suppose is that a job can be scheduled into a specific environment, but an individual job step cannot, although it is technically possible to control whether a step within a job can process on a given z/OS Image.

Most z/OS environments operate a Job Scheduling package such as TWS (was IBM's OPC), CA-7, CA-Scheduler or CONTROL-M (BMC Software), and Computer Auditors generally insist on all production work being submitted via such a product so that is a good starting point for ensuring that jobs execute on the correct system. That leaves development work which is typically submitted outside of the scheduling package from TSO/ISPF, ROSCOE (CA) or possibly even VM/CMS (IBM). In this situation there may be two slightly different requirements;

- Directing the job to a different z/OS than the one it was submitted on
- Ensuring that a particular step is only executed on a specific image.

Submitting to a specified z/OS environment

z/OS Job Control Language (JCL) provides the capability to route a job to an image other than the one it was submitted on, as do both JES2 and JES3 Job Entry Control Language (JECL). The options, which are described in the z/OS JCL Reference manual, are:

- z/OS JCL
    - `SYSAFF` parameter of the `JOB` Statement
    - `SYSTEM` parameter of the `JOB` Statement

- JES2 JECL
    - `SYSAFF` parameter of the `/*JOBPARM` Statement
    - `/*ROUTE XEQ` Statement
    - `/*XEQ` Statement

- JES3 JECL
    - `SYSTEM` parameter of the `//*MAIN` statement

Which system is the job running on?

z/OS JOB Statement parameters

Note that **SYSAFF** and **SYSTEM** are mutually exclusive that is either one could be coded but not both. The characteristics of both are subtly different but they both offer the capability to either include, or exclude environments.

SCHENV	There is an implication that this JOB Statement parameter could also be used to control the processing environment but implementation is more involved as it uses Work Load Manager (WLM).
SYSAFF	This parameter can specify up to 33 1-4 character valid JES2 member names and 8-character JES3 system names.
	A of value of * (asterisk character) indicates the system that submitted the job.
	A - (minus character) preceding a member or system name indicates that JES2 member or JES3 system is not eligible for processing the job.
	A - (minus character) preceding the first member or system name in a list indicates that **none** of members or systems in the list is eligible for processing the job.
SYSTEM	This parameter can specify up to 32 system names limited by the number of JES systems that can exist in a JESplex.
	A minus character (-) preceding the first system name in a list indicates that **none** of the systems listed are eligible for processing the job.
	For JES2, the default systems for a job are set via **SYSAFF** values that are associated with the input device contained within the JES2 Initialization Parameters..
	For JES3, the default system is the processor used for the job's class definition.

Which system is the job running on?

JES2 JECL Options

JES2 offers three possible methods of directing jobs to specific processing environments but the characteristics are different.

/*JOBPARM	Use the **SYSAFF** parameter to ensure the *conversion and execution of the job will be done on a specific system*. If **SYSAFF** is coded both processes are done on the specified system. **This means that resolution of any system symbols in the submitted JCL will be resolved on the target system, not the submission system.**
	The parameter indicates one or more JES2 member names which are eligible to process the job. The limits are the number of entries that can be coded on a **JOBPARM** statement, and the number of JES2 members that can exist in a MAS (Multi-Access Spool).
/*ROUTE	This statement can be used to send the job to a single nominated system for execution.
	The entire job is scanned for JCL errors on the input system before it is transmitted to the target system. **This means that resolution of any system symbols in the submitted JCL will be resolved on the input system, not the target system**...
/*XEQ	This statement performs the same function as the **/*ROUTE XEQ** statement.

JES3 JECL Options

There is a **SYSTEM** parameter on the **//*MAIN** Statement of JES3 which is not discussed further as most commercial organizations operate JES2 and because the use of this option seems a little more involved given the caveats in the manual. .

Which system is the job running on?

Executing a specific step on a particular z/OS environment

Whilst it is possible to direct entire jobs to specific host environment as seen by the preceding paragraphs, it is not possible to direct individual job steps to specific environments. That said it is possible to determine whether a given job step should be processed on a given host once the `JOB` has started. This ability can be accommodated by the inclusion of REXX code and use of Conditional JCL (`IF/ELSE/ENDIF`).

The key is a System Symbol called `SYSNAME` which is set in the active System Parameter Library (PARMLIB) via member `IEASYMxx` where the last two characters are installation specific as indeed will be the name of the actual library. This symbol cannot be validated directly within JCL, which is where the REXX code plays its part.

Two possible solutions are offered here as although z/OS Version 2 supports a greater range of System Symbols within Batch JCL the availability, access to them is controlled by the `SYSSYM` option within the `JOBCLASS` definition held in the JES2 Initialization Parameters. If `SYSSYM=ALLOW` is not specifically set for a job class then only a restricted set of symbols will be available and `SYSNAME` is not one of them.

Which system is the job running on?

Accessing SYSNAME with SYSSYM=DISALLOW in force

This is a complete job providing an example of how to access the system to control execution of a given job step. The REXX is coded in-stream purely for demonstration purposes.

```
//S0010    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSEXEC  DD   UNIT=SYSDA,SPACE=(TRK,(1,1,1)),LRECL=80,
//             BLKSIZE=0,RECFM=FB,DSN=&&TEMP(NAMECHK),DISP=(,PASS)
//SYSIN    DD   *
 REPRO INFILE(REXXCODE),OUTFILE(SYSEXEC)
//REXXCODE DD *,DLM='?@'
/* REXX to test &SYSNAME access capability */
thissys = MVSVAR(SYSNAME)
If thissys ¬= 'S0W2' Then
   Do
      Say thissys 'Incorrect system for job step'
      Exit 20
   End
Exit 0
?@
//S0020    EXEC PGM=IKJEFT01,PARM='NAMECHK '
//SYSEXEC  DD   DSN=&&TEMP,DISP=(OLD,DELETE)
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   DUMMY
//         IF   RC = 0 THEN
//S0030    EXEC PGM=IEFBR14
//         ENDIF
```

This job stream produced the anticipated result of bypassing the third step (S0030) as the **SYSNAME** was **S0W1** on the test system.

Which system is the job running on?

Accessing SYSNAME with SYSSYM=ALLOW in force

This is a complete job providing an example of how to access the system to control execution of a given job step. The REXX is coded instream purely for demonstration purposes.

```
//S0010    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSEXEC  DD   UNIT=SYSDA,SPACE=(TRK,(2,,1)),LRECL=80,
//    RECFM=FB,BLKSIZE=0,DISP=(,PASS),DSN=&&REXX(NAMECHK)
//SYSIN    DD   *
 REPRO INFILE(REXXCODE),OUTFILE(SYSEXEC)
//REXXCODE DD *,DLM='?@'
/* REXX to test &SYSNAME access capability */
sysname = ARG(1)
If sysname <> 'S0W2' Then Do
   Do
      Say sysname 'Incorrect system for job step'
      Exit 20
   End
Exit 0
?@
//S0020    EXEC PGM=IRXJCL,PARM='NAMECHK &SYSNAME'
//SYSEXEC  DD   DSN=&&REXX,DISP=(OLD,DELETE)
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   DUMMY
//         IF   RC = 0 THEN
//S0030    EXEC PGM=IEFBR14
//         ENDIF
```

This job stream produced the anticipated result of bypassing the third step (S0030) as the **SYSNAME** was **S0W1** on the test system.