IBM Education Assistance for z/OS V2R2

Item: Support for Symbols

Element/Component: JES3

Material current as of May 2015

Agenda

- Trademarks
- Presentation Objectives
- Overview
- Usage & Invocation
- Migration & Coexistence Considerations
- Restrictions & Limitations
- Installation
- Presentation Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.

Presentation Objectives

- In this presentation, we will:
  - Introduce how JCL symbols can be accessed during job execution
  -
  - Introduce how to use JES symbols and the JES Symbol Service (IAZSYMBL) which can be used to manage JCL and JES symbols
  -
  - Describe enhancements to the internal reader facility to allow passing JCL and JES symbols from a parent job to a submitted job
  -
  - Introduce support for substituting symbolic parameters within instream data
  -
  - Introduce ENF 78 and how it can be used by applications to request and receive notification of job events

Overview – requirements (1)

- Problem Statement / Need Addressed:
  - One requirement addressed by JES3 in V2R2 is to expand/improve usage of symbolic parameters when defining jobs to the system.
- Solution:
  - With new support it is now possible to:
    - Programmatically access JCL symbols defined in job's JCL stream at job execution-time
    - Dynamically create JES symbols via the JES Symbol Service
    - Instruct the internal reader to pass JCL and JES symbols to a submitted job. Such symbols can be used for symbol substitution in the JCL stream of the submitted job.
    - Use symbols in the instream data in the same manner as they are used in JCL stream of the job. In this way the JCL of the job and the instream data passed to the application can view the same set of symbols.

Overview – requirements (2)

- Problem Statement / Need Addressed:
  - Another requirement is to provide a JES-independent interface to identify when a job has finished execution

- Solution:
  - Application can request (by defining a special JES symbol) that JES should send notification when a particular job has completed or otherwise no longer eligible for execution.
  - This notification is provided by ENF 78

Overview – requirements (3)

- Benefits / Value:
-
    - The purpose of this line item is to provide long desired flexibility to customers in defining and managing jobs on z/OS through the use of symbols
    - Satisfies RFE 57085 / FITS MR1107134225 / SHARE SSJES313001
        - Description: JES3 should support in-stream symbolic substitution in batch JCL. This support should allow the use of symbols from various sources such as JCL symbols system symbols etc.

Overview – Topics

- Background
- EXPORT JCL statement
- JES Symbols
- JES Symbol Service (IAZSYMBL)
- Enhancements to internal reader
- Instream symbol substitution
- Job notification (ENF 78)

Background

- The symbols enhancements described in this T3 are new for JES3 at V2R2. However:
  - This functionality was originally introduced in MVS/JES2 at V2R1
  - As a result, this functionality is already documented in V2R1 manuals with 'Not supported by JES3' disclaimers
  - Such disclaimers will be removed for V2R2
- New symbols functionality for V2R2 builds on this enhancement introduced at V2R1:
  - System symbol substitution at conversion-time was extended to batch job JCL as long as the job's class is enabled with SYSSYM=ALLOW
  - Previously, system symbol substitution only occurred in Started Tasks
- For V2R2, MVS has increased the maximum system symbol table size from approximately 32KB to 56KB.
  - Also, system symbol names can be up to 16 characters

EXPORT JCL statement

- Previously, JCL symbols were used only at JCL conversion-time when the JCL stream of a job was processed
- EXPORT JCL statement is used to specify a list of JCL symbols which will be made available to the job during execution-time
- The exported symbols can be used in a number of ways:
  - Application can programmatically access the symbols using the JCL Symbol Service (IEFSJSYM) or the JES Symbol Service (IAZSYMBL)
  - The symbols can be passed to jobs submitted by the current job using the SYMLIST feature of the internal reader
  - The symbols can be used for substitution within instream data using new SYMBOLS feature of instream data sets

Usage & Invocation

- EXPORT statement syntax:

```
        //MYEXPRT  EXPORT  SYMLIST=(A,B,C, …)
```

- where A,B,C are symbol names. The symbol names in the SYMLIST have the same syntax rules as JCL symbols (without the “&” character).
- SYMLIST=* is supported.
  This form will export values of all JCL symbols set after the statement named MYEXPRT is processed.
- EXPORT statement can appear anywhere after the JOB statement.
  Note that a SET statement must be used to assign and capture the value of the symbol. EXPORT statement just makes symbol exportable – by itself it does not capture its value.

Usage & Invocation

- The scope of symbol value is job step – each step in a job may have its own unique value for the same exported symbol.
- There can be any number of EXPORT statements in the job, but the value of the exported symbol as visible during job step execution will be the last value established via a SET statement prior to or within that job step.

Usage & Invocation - Example

```
//EX1  EXPORT SYMLIST=(S1,L1)
//SET1   SET S1=STEWART,J1=JFK,N1=NIAGARA,L1=LAGUARDIA
//SET2   SET S1=SANDIEGO,F1=FRESNO
//EX2  EXPORT  SYMLIST=F1
//STEP1   EXEC PGM=USERPGM1
//STEP2   EXEC PGM=USERPGM2
//SET3   SET S1=MSP
```

- Symbols S1, L1 and F1 are exported.
- Symbol S1 will have value SANDIEGO in STEP1 – because this was the last value prior to and within STEP1
- Symbol S1 will have value MSP in STEP2 – because its value was changed within STEP2
- Symbol F1 will have a null value because its value was never SET after its EXPORT statement

JES Symbols

- Managed and accessed via JES Symbol service (IAZSYMBL)
- JES symbols can be:
  - Used to communicate between applications running within the same job step
  - Passed as JCL symbols to jobs submitted through internal reader (SYMLIST feature)
  - Used for substitution within instream data sets (SYMBOLS feature)
  - Used to communicate between application and JES internal reader (via a set of special predefined JES symbols)
- JES Symbols are very similar to JCL symbols with relaxed restrictions:
  - Longer name – up to 16 characters
  - Longer value – up to 4096 bytes

JES Symbol Service (IAZSYMBL) (1)

- JES Symbol Service (IAZSYMBL) is used to manage JES symbols (create/update/delete/extract) and to provide seamless read-only access to JCL symbols made available via EXPORT statement
- JES Symbol service is an interface which is key to several enhancements discussed elsewhere in this presentation:
  - Symbols created dynamically by the JES Symbol Service can then be passed to submitted jobs
  - Interacting with internal reader to request special processing for a submitted job
- A related symbol service is the JCL Symbol Service (IEFSJSYM) which manages JCL symbols

JES Symbol Service (IAZSYMBL) (2)

- JES Symbol Service (IAZSYMBL) is used to manage JES symbols (create/update/delete) and to provide seamless read-only access to JCL symbols made available via EXPORT statement
- JES Symbols can be created on a task level or on a job step level
- JES Symbol overrides JCL symbol with the same name
- JES Symbol on a task level overrides JES symbol with the same name on a job step level
- JES Symbol service is an interface which is key to several enhancements discussed elsewhere in this presentation:
  - Symbols created dynamically by the JES Symbol Service can then be passed to submitted jobs
  - Interacting with internal reader to request special processing for a submitted job

JES Symbol Service (IAZSYMBL) (3)

- JES Symbol Service (IAZSYMBL) supports the following requests:
  - CREATE – create one or more JES symbols and assign initial values
  - UPDATE – update values of selected JES symbols
  - DELETE – delete JES symbols and their values
  - EXTRACT – return values of selected JES symbols
- JES Symbol service supports operations over multiple JES symbols at a time (including wildcard support for EXTRACT and DELETE requests)
- Note that EXTRACT operation searches for requested symbols in this order:
  - Task level JES symbols
  - Job step level JES symbols
  - JCL symbols exported by EXPORT statement

Usage & Invocation - JES Symbol Service (IAZSYMBL)

- Interface to JES Symbol Service includes two macros
  - IAZSYMBL – invocation macro
  - IAZSYMDF – data definition macro which contains declarations for the parameter list and various structures used by the service
- Similar in style to SSI interface
- Key data structures used by the JES Symbol Service:
  - Parameter structure
    - Filters, options, return and reason codes, feedback data, etc
  - JES Symbol table structure which is used
    - For input – to create and update JES symbols
    - For output – to return symbol information

Enhancements to internal reader - symbols

- JES3 allows passing symbols from a parent job to a submitted job
  - These symbols are available for substitution during JCL processing in the same manner as symbols created via the SET statement
  - Such symbols are implicitly exported within the submitted job – EXPORT statement is not required
  - Parent job and submitted job can now use consistent set of symbols
-
- The following symbols can be passed to internal reader:
  - JCL symbols available to a parent job
  - Dynamically created JES symbols as long as they conform to JCL requirements

Usage & Invocation - enhancements to internal reader - symbols

- Syntax for statically allocated internal reader:

```
//SYSUT2   DD SYSOUT=(A,INTRDR),SYMLIST=(A,B,C)
```

- where A,B,C in the SYMLIST are symbol names. The symbol names in the SYMLIST have the same syntax rules as JCL symbols (without the “&” character).
- SYMLIST=* is supported.
  This form will pass all exported JCL symbols and all defined JES symbols which conform to JCL symbol rules to the submitted job
- SYMLIST keyword also supported on TSO ALLOCATE command
- Dynamically allocated internal reader has equivalent function via text unit DALSYML (TU key X'802B')

Enhancements to internal reader - feedback

- New type of programmatic feedback was added to internal reader
-
- When job submitted through internal reader is successfully accepted by JES (job submission succeeded) the following JES symbol is set at the task level:
  - SYS_LASTJOBID
    The value of this symbol is 8-character JES job identifier of the job which was just submitted
  -
- When job submission fails, internal reader will set this symbol to null value

Usage & Invocation - enhancements to internal reader - feedback

- To obtain feedback from the internal reader, an application can use the JES Symbol Service (IAZSYMBL) to EXTRACT the value of the feedback symbol
  - SYS_LASTJOBID

- Note that a call to JES Symbol Service must be made in the same task which wrote JCL stream for the submitted job into internal reader.

Instream symbol substitution (1)

- Symbols within the records of instream data sets are used in much the same manner as symbols are used in the JCL
- Symbol substitution occurs when an application reads a record from the instream data set
- The following types of symbols can be used:
  - JCL symbols
    - made available at job execution-time through the use of EXPORT statement
  - JES symbols
    - dynamically created by application using JES Symbol Service (IAZSYMBL)
  - System symbols
    - MVS system symbols associated with either the system where the job executes, or the system where the job undergoes conversion

Instream symbol substitution (2)

- This feature allows application designers to use the same symbolic parameters in both a job's JCL and its instream data sets

- Instream symbol substitution is optional and is controlled by the SYMBOLS keyword on DD JCL statement which defines the instream data set. The default is not to perform any symbol substitution.

- Optionally, diagnostic logging of the real-time substitution can be requested. This is controlled by the second optional value of the SYMBOLS keyword.

Usage & Invocation - instream symbol substitution

- SYMBOLS keyword has the following format
  SYMBOLS=[( ] JCLONLY | EXECSYS | CNVTSYS [, DDname)]

- SYMBOLS=JCLONLY
  Substitute JCL symbols made available by the EXPORT statement and JES symbols dynamically created via JES Symbol Service (IAZSYMBL)

- SYMBOLS=EXECSYS
  Perform substitution as for JCLONLY. In addition, substitute system symbols from the system where this job executes

- SYMBOLS=CNVTSYS
  Same as SYMBOLS=EXECSYS except that system symbols from the conversion system are used.

- DDname is an optional DD name (which must be provided by the user) which will receive the symbol substitution log.

Usage & Invocation - instream symbol substitution - example

```
//   EXPORT SYMLIST=(DSN,VOL)
//   SET DSN='ABC.DATA',VOL='123456'
//STEP1   EXEC PGM=USERPGM1
//DATA    DD  DSN=&DSN,DISP=SHR
//SYSIN   DD  *,SYMBOLS=EXECSYS
   SYSTEM=&SYSNAME,DSNAME=&DSN,VOLUME=&VOL
   FUNCTION='&APPL_NAME'
/*
```

- After reading record from SYSIN data set on system SY1, application will see this:

```
SYSTEM=SY1,DSNAME=ABC.DATA,VOLUME=123456
FUNCTION='RECORD SEARCH'
```

- This example assumes that before reading from SYSIN data set, application has created JES symbol APPL_NAME with the value 'RECORD SEARCH'.

Usage & Invocation – substitution logging - example

```
//MYLOG    DD   SYSOUT=A
//SYSIN    DD   *,SYMBOLS=(EXECSYS,MYLOG)
SYSTEM=&SYSNAME,DSNAME=&DSN,VOLUME=&VOL
.  .  .
```

- The MYLOG data set will show results of the substitution

```
SYSIN     : RECORD 1 BEFORE SUBSTITUTION
SYSIN     : SYSTEM=&SYSNAME,DSNAME=&DSN,VOLUME=&VOL
SYSIN     : RECORD 1 AFTER SUBSTITUTION
SYSIN     : SYSTEM=SY1,DSNAME=ABC.DATA,VOLUME=123456
```

Usage & Invocation – composite example

```
JCLPARNT.TEST.JCL:
          //JCLPARNT JOB MSGLEVEL=(1,1),MSGCLASS=A,CLASS=B
          //          EXPORT SYMLIST=(DSNAME,VOLSER)
          //MYSET1    SET DSNAME=JCLCHILD.TEST.JCL,VOLSER=D98765
          //STEP1     EXEC PGM=IEBGENER
          //SYSIN     DD   DUMMY
          //SYSPRINT DD   SYSOUT=*
          //SYSUT2    DD   SYSOUT=(,INTRDR),SYMLIST=(DSNAME,VOLSER)
          //SYSUT1    DD   DSN=&DSNAME,VOL=SER=&VOLSER,UNIT=3390,DISP=SHR

JCLCHILD.TEST.JCL:
          //JCLCHILD JOB MSGLEVEL=(1,1),MSGCLASS=A,CLASS=S12
          //STEP1     EXEC PGM=IEBGENER
          //SYSIN     DD   DUMMY
          //SYSPRINT DD   SYSOUT=*
          //SYSUT2    DD   SYSOUT=*
          //SYSUT1    DD *,SYMBOLS=JCLONLY
           &DSNAME. AND &VOLSER. FROM PARENT JOB ARE USED
          /*
```

- After submitting the parent JCL, the child JCL is submitted through the INTRDR. During execution of the submitted job, the instream records are copied to SYSOUT (SYSUT2) which will have this content:

```
          JCLCHILD.TEST.JCL AND D98765 FROM PARENT JOB ARE USED
```

Job notification (ENF 78)

- This feature allows an application to request special job tracking and instructs JES3 to send notification when a job is no longer eligible for execution.

- Job Notification is sent via ENF 78

- Application can associate an optional data string along with the notification request. This data string can be up to 4096 bytes long and will be included by JES3 in the notification record (ENF 78).

Usage & Invocation – job notification

- To request job notification, application must use the JES Symbol Service to create a special symbol named SYS_JOB_NOTIFY before submitting the JCL stream of the job through the internal reader.

- Value of the SYS_JOB_NOTIFY symbol becomes a data string which is included in ENF 78 when signalled to the parent job.

Migration & Coexistence Considerations

- Symbols functionality has not been rolled back to previous releases.
- JES3 Global must be at V2R2 at all times.
- Job JCL using new symbols functionality must undergo C/I processing at V2R2 level. JES3 will steer jobs to V2R2 C/I only after symbols usage is detected.
- Parent job using the SYMLIST feature of the internal reader, must execute on a V2R2 system for symbols to be passed to submitted job.
- When the following features are used, job becomes ineligible for processing on down-level systems in the JESplex:
  - Submitted job is passed symbols from a parent job using the SYMLIST feature of the internal reader
  - SYMBOLS feature is specified for an instream data set (assumes conversion of job is performed on a V2R2 system)
  - Job notification function

Restrictions & Limitations

- Symbols associated with the job are lost when:
  - Job is transmitted via NJE to a remote node
  - Dump job facility is used to dump a job to tape

Installation

- JES3 release toleration APAR OA43563 is required for systems at V2R1 or V1R13 in the JESplex

Presentation Summary

- New JES3 enhancements to job symbol processing will improve flexibility when creating and maintaining JCL.
- A consistent set of symbols can be used in multiple jobs by allowing job to access its JCL symbols at execution-time and by allowing parent jobs to pass their set of symbols to the submitted jobs.
- Now the same symbols used in JCL of the jobs can be used in instream data sets.
- It is expected that application programmers as well as system programmers will benefit from these changes.

Appendix

- Publications
  - *z/OS V2R2.0 JES Application Programming* – SA32-0987
  - *z/OS V2R2.0 JES3 Messages* – SA32-1007
  - *z/OS V2R2.0 JES3 Commands* – SA32-1008
  - *z/OS V2R2.0 MVS JCL User's Guide* – SA23-1384
  - *z/OS V2R2.0 MVS JCL Reference* – SA23-1385