5 Tips for developing JCL Standards

Standards are required to effectively manage production Job Control Language (JCL). A majority of data centers have embraced the concept. They realize the performance and productivity benefits of implementing JCL standards for naming conventions (JOBS, PROCs, programs and data sets), allocating data sets (force the use of SMS), structuring JCL statements (keywords in a specific order, and one keyword per line), and validating site-specific parameters (accounting field, job class, SYSOUT class, etc.).

Although your site-specific results will vary, one company reported a ten-fold savings over the cost of implementing their JCL standards project.¹

It’s important to take time to define detailed standards. Analyze site-specific requirements and your JCL asset. Who is your internal customer? What is their bottom-line requirement?

You can have a great list of standards, but if they’re not enforced, you won’t realize the benefit. JCL Standards can be enforced manually or automatically. If your goal is to have a standardized and error-free JCL environment, we strongly encourage you to consider automatic enforcement of JCL standards due to the sheer volume of JCL members and statements within those members.

This paper offers general tips for creating standards and encourages periodic reviews of your JCL standards.

Topical list of subjects covered in this paper:
- AN OVERVIEW OF JCL STANDARDS
- THE FOUR CATEGORIES OF JCL STANDARDS
- FIVE TIPS FOR THE JCL STANDARDS DEVELOPER
- ABOUT DIVERSIFIED SOFTWARE
- APPENDIX A: JCL STANDARD RESOURCES

To What Degree is Your JCL Standardized?

In a recent survey of mainframe users conducted by Diversified Software, 95% of respondents recognized the need for their JCL to be standardized.

Source: Mainframe User Survey
March 15, 2005

AN OVERVIEW OF JCL STANDARDS ENFORCEMENT:

The purpose for JCL standards is to establish uniformity of job control language coding. And the best place to start is when the JCL is first created.

Typically, there are four phases of the JCL Lifecycle: Maintenance, Test, Stage, and Production. The Application Development group creates and maintains the application programs and their associated JCL during the “Maintenance” phase. Depending upon site-specific processes, the application programs and JCL are then tested/validated during the “Test” and “Stage” phases (some shops may combine these phases). Finally, the programs and JCL move into the “Production” libraries. Standards (including JCL format) should be enforced in each of these phases.

Changes to the JCL are inevitable and the JCL Lifecycle starts over. Application Development checks-out the JCL member(s) from the Production library, JCL changes are made, the JCL member(s) are validated, site-standards are enforced, and the modified JCL is promoted into the Production library.

The Production JCL used to run the batch processing must be error-free. Preventable errors waste time and resources. That’s why it is important to manage the JCL asset. So JCL validation and standards enforcement should be done not only as part of the Promotion process, but also prior to a Production run.

JCL-related Activities in Today’s Data Center

An analysis of the JCL-related activities in today’s data center reveals some interesting findings:

- The amount of JCL running batch production is about the same as it has always been.
- There is as much maintenance performed on JCL as before.
- The amount of new JCL written/created is decreasing.
- There is a high level of activity in maintaining, standardizing, converting, and cleaning-up of the JCL environment because:
  - JCL has been ignored for too long, and it has deteriorated (obsolete parameters, etc.).
  - System resources are being wasted (allocation of unreferenced libraries, etc.).
  - Maintenance has become difficult because of non-standardization.
  - There is a proliferation of duplicates and near duplicates.
- There is an ever-decreasing batch window that mandates efficient operations (error-free JCL, etc.).

Source: Mainframe User Survey
March 15, 2005
THE FOUR CATEGORIES OF JCL STANDARDS:

Most of the JCL Standards will fall into one of the following broad categories:

- Format and readability
- Naming conventions
- Performance enhancers
- Site-specific requirements

**Format and readability:** this category addresses the appearance and the ease with which the JCL worker can find the item of interest and take action. These are probably the easiest Standards to identify and fix. The most common of these include placement of a single keyword per line (and in a consistent order), and column alignment for operands and keywords.

Reformat example (these examples are for the purpose of discussion only, and may not apply to your site):

- **Example for all JCL statements:**
  - Align all operands in column 12.

- **Example for DD statements only:**
  - Align parameters in column 16.
  - Place only one parameter per line.
  - Order the keywords as follows: DSN, DISP, UNIT, VOL, SPACE

Sample JCL before reformatting (left). Multiple keywords per line and misalignment make it difficult to read.
Sample JCL after reformatting (right). Single keywords per line and column alignment make it easier to read.
**Naming conventions:** this category enhances the ability to “make sense” out of JCL. For example, just by looking at the data set name at one site, programmers and operators should be able to tell that it was created by the Accounts Receivable system (even though it’s referenced in a JCL member that is part of the General Ledger system). These Standards are more difficult to define as they may affect many different departments. The most common naming conventions include the Jobname and data set name.

Naming convention example (these examples are for the purpose of discussion only, and may not apply to your site):

- **Example for Jobname naming conventions:**
  - The first two characters of the Jobname indicate the system (i.e. “GL” is General Ledger, “AR” is Accounts Receivable, etc.).
  - The next four characters of the Jobname indicate the environment (i.e. “PROD” is the Production environment, “TEST” is the Test environment).
  - The final two characters indicate a unique number associated with the job.
  - The JCL Library PDS member name must match the Jobname specified within the JCL member.

- **Example for data set naming conventions:**
  - High level qualifier indicates the environment (i.e. “DEV”, “TEST”, “STAGE”, or “PROD”).
  - The 2nd node indicates the first two characters of the system (i.e. “GL” is General Ledger, “AR” is Accounts Receivable, etc.).
  - If four nodes are specified, the 3rd node is either “CURRENT”, “ARCHIVE”, or “REPORT”.
  - The final node is “DATA”, “CNTL”, “LOAD”, or “PROCLIB”.

```
---+----1----+----2----+----3----+----4----+----5----+----6----+----7--
//GLPROD01 JOB (37489)
//JS020 EXEC PGM=IEWL
//DD1 DD DSN=PROD.AR.CURRENT.DATA,
  // DISP=SHR
//DD2 DD DSN=PROD.GL.REPORT.DATA,
  // DISP=NEW,
  // SPACE=(CYL,(21,1)) ,
  // BLKSIZE=2048
```

**Jobname naming convention.** “GLPROD01” indicates this is a General Ledger Production job.

**Data set naming conventions.** “PROD.AR…” indicates DD1 is an Accounts Receivable Production data set.

“PROD.GL…” indicates DD2 is a General Ledger Production data set.
**Performance enhancers:** this category reduces opportunities for throughput bottlenecks and wasted system resources. Again, these Standards are difficult to define as they may affect many different departments. The most common conventions include JCLLIB concatenations, PROC definitions/concatenations, BLKSIZE values, and SMS.

Performance Standards example (these examples are for the purpose of discussion only, and may not apply to your site):

- **Example for PROC Standards:**
  - Do not use in-stream PROCs (must use Cataloged PROCs).
  - Do not use more than 2-levels of Nested PROCs.
  - Use the standard JCLLIB library concatenation (ensure consistency and make sure obsolete libraries are not specified).

- **Example for BLKSIZE Standards:**
  - If specified, the BLKSIZE value must be Zero (0).
  - JCL created prior to MVS ESA might contain a value other than Zero. With ESA and higher releases, let the system determine the block size. If the BLKSIZE value is other than Zero, it should be changed to Zero and a warning message should be issued.
  - It is OK if the BLKSIZE keyword is missing.

- **Example for SMS Standards:**
  - Use SMS to allocate all new data sets.
  - Do not use SPACE, UNIT, and VOL=SER keywords (want to take advantage of SMS). If these are specified, issue an error message and do not promote into the Production JCL Library.

---

```
//GLPROD03 JOB (37489)
// JCLLIB ORDER=(PROD.GL.PROCLIB,PROD.AR.PROD.PROCLIB,
// IBM.SYSTEM.PROCLIB)
// JS023  EXEC  GLPROC23
//PSTEP02.DD1 DD  DSN=PROD.AR.CURRENT.DATA,
//  DISP=SHR
//PSTEP02.DD2 DD  DSN=PROD.GL.REPORT.DATA,
//  DISP=NEW,
//  SPACE=(CYL,(21,1)),
//  BLKSIZE=2048
```

**PROC Standards.** Specifies Cataloged PROC GLPROC23 and the standard JCLLIB library concatenation.

**BLKSIZE Standards.** DD2’s “BLKSIZE=2048” keyword should be changed to “BLKSIZE=0”.

**SMS Standards.** DD2’s “SPACE” keyword should force an error message. The DATACLAS keyword and appropriate values should be used to allocate this data set.
**Site-specific requirements:** this category verifies site-specific keyword and parameter values that ensure the job will run as expected. These Standards address site-specific environment and policy decisions. The most common conventions include Accounting Field parameters, Job CLASS values, and SYSOUT class values.

Site-specific convention Standards example (these examples are for the purpose of discussion only, and may not apply to your site):

- **Example for Accounting Field parameters:**
  - Specify the Accounting Field following the standard format and matching one of twelve valid accounting values. The accounting value specified must match the appropriate system as indicated by Jobname (i.e. a General Ledger job must specify a valid General Ledger accounting number).

- **Example for Job CLASS Standards:**
  - Specify an appropriate job CLASS.
  - Specify appropriate Job CLASS values: “T” for jobs requiring Tapes or Tape Cartridges, “Q” for jobs running less than 0.5 CPU seconds, and “P” for all other Production jobs.

- **Example for SYSOUT Standards:**
  - Specify appropriate SYSOUT classes: “A” for jobs printing 5,000 pages or more, “B” when specifying forms, and “C” for all other Production jobs.
  - Do not specify more than 10 copies for a SYSOUT. If more copies are needed, route to SYSOUT class “X”, specify “TYPRUN=HOLD”, and notify the operator of this special request.

- **Example of other miscellaneous Standards:**
  - Use step name sequencing (use STEP010, STEP020, STEP030 ... as opposed to stepthis, stepthat, ...). This allows for better readability, and can accommodate future expansion (i.e. you can easily insert a new STEP015 between STEP010 and STEP020).
  - Use a “//**” comment block for each step describing what the step does. Keep it short and up-to-date.
  - Don't use &&temporary data sets. Create a data set, use it, and then delete it at the end of the job.
  - Don’t use tape files for sorts.
  - Don’t use //SORTWK DDs (no longer necessary). Let system take care of these allocations for you.
Accounting Field Standards. The value “37489” is an acceptable value for General Ledger batch jobs.

Job CLASS Standards. “CLASS=P” is an acceptable value for General Ledger batch jobs.

SYSOUT Standards. “A” and “J” are acceptable SYSOUT values. Also, copies are less than 10.
FIVE TIPS FOR THE JCL STANDARDS DEVELOPER:

Here is a list of five simple tips that can help JCL Standards developers tackle their JCL environment.

Tip #1. Define your JCL Standards. Since you want to cover all the requirements for JCL to run within your data center environment, take the time to carefully document all your standards. This could be a challenge because some of the standards may span across several departmental lines. Make sure that the standards are reasonable, enforceable, and allow room for future growth. It may be easier to divide-and-conquer the “standards beast” by addressing one application, one department, or one division at a time.

Tip #2. Distinguish between “Standards” and “Guidelines”. As you’re creating your list of JCL Standards, you should categorize the requirements as either a Standard or a Guideline:

- Standards are those rules that MUST be followed. For example, your site may designate “data set naming conventions” as a “Standard”. In other words, you MUST follow the data set naming convention or your JCL will not be promoted into the Production JCL Library.

- Guidelines are those rules that SHOULD be followed, but are not enforced. For example, your site may designate the rule “Don’t use && temporary data sets…” as a “Guideline”. In other words, it is recommended that you SHOULD NOT use &&temporary data sets. If you do use them, you may be issued a warning message (but it would not prevent your JCL from being promoted into the Production JCL Library).

- Over time, it may be appropriate to migrate a “Guideline” into the “Standards” category.

Tip #3. Enforce your “Standards” using an automated software product. Although enforcement of these standards can be manual or automated, most data centers choose to automatically enforce the standards for these reasons:

- Time – ensuring that all jobs will be checked within a reasonable amount of time (most data centers manage thousands of JCL members).

- Accuracy – ensuring that all of the standards are correctly enforced.

- Accountability – ensuring that every standard is applied.

Make sure that you program for all conditions:

- Many standards to enforce – make sure all are covered.

- Assign different levels of enforcement (severity codes) for “Standards” vs. Guidelines”.

- Include error handling code (when parameters are missing or incorrect).

- Automatically correct the more common errors (using an intelligent change feature).
Tip #4. Develop test cases. When automatically enforcing the standards for thousands of jobs, it’s important to develop a set of test cases for these reasons:

- Validate the standards program (make sure it’s catching the standards violations).
- Validate the “Standards” and “Guidelines” (make sure it’s distinguishing between the MUSTs and SHOULD DOs).
- Validate the error handling code (make sure it can handle problem conditions).
- Validate the intelligent change code (make sure that the common errors will be corrected).

Tip #5. Maintain your “Standards”. Just as JCL changes over its lifecycle, so will JCL Standards. Your standards may change due to new operating system features, company policy, legal policies, regulatory policies, corporate mergers, or perhaps event performance issues. Make it a point to perform a periodic review of your JCL Standards.

Here are a couple examples of why a periodic review might be important to your data center:

- The JCL Environment evolves with new requirements and updated operating systems. MVS ESA introduced the optimized parameter “BLKSIZE=0”. In order to take advantage of this new feature, most data centers used Intelligent JCL Change and JCL Standards programs to implement global changes and enforce the usage of this parameter.
- Introduction of SMS moved some standards out of the JCL and into the SMS realm. In order to take advantage of SMS, most data centers used Intelligent JCL Change and JCL Standards programs to implement global changes and enforce the usage of SMS (prohibiting allocations of data sets within the JCL).
- As noted earlier, sometimes it may be appropriate to migrate a “Guideline” into the “Standards” category. Perhaps use of SMS for data set allocations was initially a “Guideline”, but now management has decided it should be a “Standard”. In order to migrate this “Guideline” into a “Standard”, you could use Intelligent JCL Change and JCL Standards programs to implement global changes and enforce the usage of SMS (prohibiting allocations of data sets within the JCL).
- A corporate merger includes data center consolidation. This may require an addition of new “Standards” and “Guidelines” that must co-exist with the old “Standards” and “Guidelines”, or perhaps modifications of existing “Standards” and “Guidelines” to accommodate the new corporate data.
Conclusion: There is an old adage that says, “If you are not enforcing your standards, then you don’t have standards!” This statement is just as true today as it was when it was first stated.

In the spirit of “enforcing standards”, many sites have put hooks in the system to “foil” jobs as they are submitted and processed by JES and/or different system exits. While absolutely effective, the problem with this approach is that it is too late in the process and causes a production job failure. The crux of effective standards enforcement is to make sure that the JCL conforms with the standards before they become a problem.

No matter how “good” your JCL Standards are at the time of creation or how often they are reviewed throughout its lifecycle, JCL Standards can quickly become outdated. Standards can include format, naming conventions, performance, and other site-specific rules. Your JCL may be running error-free now, but it may not be running as efficiently as it once did. These problems may not happen overnight, but could creep in and have a cumulative negative impact on the operational efficiency of the JCL.

“Unauthorized” changes may end up in Production JCL libraries when “emergency patches” are applied, pre-existing and non-standard JCL is “grandfathered” in, and exceptions are granted due to various reasons. Therefore you should consider undertaking a periodic “JCL Review” of your JCL environment.

So how would you know if the Production JCL is well managed? Here are a few indicators:

- Zero Production JCL errors (JCL is validated prior to the production run).
- All JCL is standard (follows all JCL Standards).
- All JCL is efficient (no obsolete parameters or unused library references are found in the JCL).
- JCL Standards are published (and automatically enforced).
- No unnecessary PROCs or PARM members are found in the JCL.
- Fast, efficient Test to Production turnover process (automated JCL validation and intelligent change).
- Controlled JCL Lifecycle (from creation through maintenance).

Do You Have Plans for JCL-specific Projects in the Near Future?

In a recent survey of mainframe users conducted by Diversified Software, 57% of respondents said they have “on-going” or “planned” JCL-specific projects. 20% of those were JCL Standardization projects.

Source: Mainframe User Survey
March 15, 2005
ABOUT DIVERSIFIED SOFTWARE:

Since 1978, Diversified Software has helped customers effectively manage the technology, people and processes used in the operation of their batch applications. The majority of the world's largest and most successful data centers depend on Diversified Software advanced products and JCL-specific professional services to effectively deploy and operate their mission-critical business applications. Diversified Software products and services have been implemented in over 30 countries worldwide.

JCL Asset Management
- **PRO/JCL®** - Designed to help achieve and maintain an error-free, standardized and optimized z/OS and OS/390 production JCL lifecycle. Designed for the most sophisticated data centers.
- **PRO/JCL® Workstation** - The optional Windows component of PRO/JCL, based on our most advanced technology. PRO/JCL Workstation makes JCL more intuitive and efficient than ever before.
- **JOB/SCAN™** - The most widely used JCL Validation and JCL Change product, designed to support error-free mission-critical production JCL. Designed to be useful for any data center.

Cross Reference and Documentation
- **INFO/X® Enterprise** – Automatically captures, cross-references and makes available up-to-date information about production components and relationships. Designed for the most sophisticated data centers.

Diversified Software Services
- Diversified Software provides best of breed technology and a team of IT experts experienced in analyzing and refurbishing your entire JCL lifecycle process. By utilizing IT consultants with specific experience in JCL Lifecycle Best Practices, organizations can get results much more quickly than utilizing internal resources alone.

Diversified Software Systems, Inc., 18635 Sutter Blvd., Morgan Hill, CA 95037

**Phone:** 408.778.9914  **Email:** info@diversifiedsoftware.com  **Web:** www.diversifiedsoftware.com

Diversified Software is proud to be an IBM Business Partner.

© 2005, Diversified Software Systems, Inc. All rights reserved.
INFO/X, DOCU/TEXT, PRO/JCL, and JOB/SCAN are trademarks and registered trademarks of Diversified Software Systems, Inc. Other referenced product names are trademarks of their respective holders.
APPENDIX A: JCL STANDARD RESOURCES

Listed below are excellent resources for further study about JCL Standards. To obtain copies of the white papers or articles, or to find out more information about the training classes, contact Debbie Boys at 408.778.9914 x257 or by email at dboys@diversifiedsoftware.com

**White Papers**

- “Sarbanes-Oxley & Your JCL” – This White Paper offers specific examples for using Diversified Software products for some of the key areas that are affected by a SOX compliance project.

- “Overcoming Multiple LPAR Validation Challenges” – A discussion of the challenges of and solutions for validating JCL in an environment with multiple LPARs (Test, Production, etc.).

- “Ensure that Your JCL Batch Production is Protected” – This is a white paper that describes the key considerations for preparing your batch environment for a Disaster Recovery drill using DOCU/TEXT and JOB/SCAN. PRO/JCL and INFO/X are equally applicable.

- “JCL Best Practices” – A discussion of industry accepted best practices for managing JCL effectively.

**Articles**

- “What Makes Good JCL Format?” – Olivia R. Carmandi, MVS Training

- “Gain Productivity By Creating and Implementing a JCL ‘Production Checklist™’” – Olivia R. Carmandi, MVS Training

- “Automating the Enforcement of JCL Standards” – AFCOM Communique - January 2004


**Training opportunities**

- **JCL Standards** – Diversified Software monthly product training classes include instruction on the automation of JCL Standards. These classes are taught at the Diversified Software Colorado Springs training center, and are available at no charge to Diversified Software customers (as part of their maintenance agreement). For more information, contact us at 800.273.3774 (719.265.6300) or visit the web at www.diversifiedsoftware.com/ccare/training.html

- **JCL Basics** – MVS Training will be conducting training classes May, 2005, July, 2005, and October, 2005, at the Diversified Software Colorado Springs training center. For more details (including fees and registration information), contact MVS Training at 800.356.9093 (412.269.9668) or visit the web at www.mvs-training.com/jclpublic.asp