Redbooks

ibm.com/redbooks

A Guide to JES3 to JES2 Migration

Luiz Fadel

Lutz Kuehner

Ricardo Paranhos

IBM Z

IBM

Redbooks

International Technical Support Organization

A Guide to JES3 to JES2 Migration

November 2019

SG24-8427-01

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

Second Edition (November 2019)

This edition applies to version 2 release 4 of IBM z/OS (product number 5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

CICS®
Db2®
GDPS®
IBM®
IBM Z®
IBM z Systems®
IBM z14®
MVS™
Parallel Sysplex®
RACF®
Redbooks®
Redbooks (logo) ®
System z®
Tivoli®
VTAM®
z Systems®
z/OS®
z13®

The following terms are trademarks of other companies:

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redbooks® publication provides information to help clients that have JES3 and want to migrate to JES2. It provides a comprehensive list of the differences between the two job entry subsystems and provides information to help you determine the migration effort and actions.

This book considers the features of JES2 as available on releases of IBM z/OS® V2R3 and V2R4. It should be used with *JES3 to JES2 Migration Considerations*, SG24-8083.

This publication is divided into three parts:

- Part 1, “Planning to migrate from JES3 to JES2” on page 1, gives you information to make the decision and plan your migration.
- Part 2, “Use case study” on page 111, provides a Use Case Study that is based on an actual customer experience in a successful migration.
- Part 3, “Appendixes” on page 193, provides an appendix with sample tools that can help the migration process and exploitation of some of the new JES2 functions.

This book is aimed at operations personnel, system programmers, and application developers.

Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Luiz Fadel is a consultant for Maffei Consultoria em Informatica Ltda. He retired from IBM in 2013 as an IBM Distinguished Engineer, responsible for supporting IBM System z® for the Latin America region, part of the Growth Markets Unit. He joined IBM in 1969 and has supported large systems since then, including working on two assignments with the International Technical Support Organization (ITSO). He was a member of the zChampions team and the co-author of several IBM Redbooks publications.

Lutz Kuehner is a Senior z/OS System Engineer at Credit Suisse in Switzerland. He has 33 years of experience in IBM z Systems®. Lutz has worked for 10 years in the IBM Z® Presales support in Germany, developing and providing professional services for customers in the financial market. In addition to co-authoring several IBM Redbooks publications since 2001, he has been an official ITSO presenter at ITSO workshops.

Ricardo Paranhos is a Senior z/OS System Programmer at IBM Brazil. He has 30 years of experience in IBM z Systems acting as Solution Architect for z/OS projects and z/OS support. He holds a degree in Electrical Engineer and Bacherol of System Analysis. He is a member of IBM Academic Initiative program since 2014. His areas of expertise include z/OS System Performance and Capacity Planning, IBM Parallel Sysplex® environment, DFSMS, and Legacy Modernization. He has developed some utilities for users that are available on the CBTTAPE site.

Thanks to the authors of the previous editions of this book:

- Luiz Fadel
- Nobuhiko Furuya
- Lutz Kuehner
- Ricardo Paranhos

Thanks to the following people for their contributions to this project:

Lydia Parziale, Project Manager
International Technical Support Organization, Poughkeepsie Center

Tom Wasik, Gary Puchkoff
IBM USA

Yves Colliard
YCOS GmbH Germany

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Part 1

Planning to migrate from JES3 to JES2

This book provides information for organizations that have JES3 and JES2 and want to consolidate onto JES2. This publication is also beneficial to organizations that have only JES3 and are considering migrating to JES2.

In this part, we cover the information that you need to help plan and manage your own migration.

Perhaps one of your first questions is: “Why would an enterprise want to convert to a different job entry subsystem”? You might be considering such a move for the following reasons:

- JES3 has few new functions; JES2 features many enhancements.
- You might have both JES3 and JES2 and want to have consistent JCL and procedures across all your z/OS systems.
- Because more JES2 installations exist than JES3 installations, it might be easier in your area to find personnel with JES2 experience.
- It is possible that products you want to use do not support JES3.
- Perhaps a certain product is better tested with JES2.
- You might find that JES3 includes features that you no longer use.
- New functions often appear in JES2 before they appear in JES3, and you want to remain current in your product levels.
- You performed a financial analysis and found that costs might be reduced by consolidating systems and converting them to JES2.
- You are working to improve your availability and JES2 appears to provide more flexibility to make dynamic changes than JES3.

Not all of these reasons apply in every case. As with any IT strategy, your decision is based on a thorough analysis of the costs and benefits of migrating.

This book helps you identify what the migration effort entails. For some JES3 installations, the migration might be relatively easy, but for others, it is time-consuming and complex. It depends on the extent to which you use the capabilities that are unique to JES3.

Positioning for migration

Many of the issues that must be addressed when performing a JES3-to-JES2 migration pertain to the use of facilities that, at one time, were provided by JES3 only. Over time, many of these facilities were provided by the operating system. However, people continue using things with which they are familiar.

Even if you are not considering migrating to JES2, it is a good idea to ensure that any *new* applications or new jobs avoid the use of facilities that are unique to a particular Job Entry Subsystem (JES).

Most enterprises take several years to deliberate over whether they perform the migration. During that time, you might be creating many more issues that then must be addressed as part of the migration.

In the opinion of the authors, it is a good investment of your time to put tools, documentation, and education in place now to ensure that your users (including operators, production schedulers, application developers, and system programmers) stop using mechanisms that are unique to JES3. If you have JES2 and JES3 today, it is worth considering to stop using mechanisms that are unique to any JES.

We highlight changes throughout this book that can be made now that will not affect current operations under JES3, but that make the migration easier if you decide to go down that path.

1

Why JES2?

As of the time of this writing, IBM announced its directions related to JES3 future. For more information, see this web page.

This chapter presents arguments for technical professionals and managers about why JES2 is a better option than JES3 not only because of financial issues (JES3 requires another license fee and JES2 does not), but because most JES3 customers successfully migrated to JES2. It also shows that the migration process can be easy and low risk.

This chapter includes the following topics:

1.1 Introduction

The decision to migrate from JES3 to JES2 can be difficult because of the following factors:

- The current staff lacks JES2 skills.
- The cost of migration exceeds the reduction in license fee.
- High dependency on JES3 features that you cannot give up.
- Your organization sees few benefits in a JES2 migration.

Additionally, some customers face the following questions:

- Is JES3 going away?

  IBM announced the JES3 withdraw after the release following the z/OS 2.4 release. IBM remains committed to help the JES3 clients successfully migrate to JES2 and continues to be available to field questions and help clients plan for a successful migration.

- Will JES3 feature new functions?

  Unlikely. However, required changes to maintain the current functionality of JES3 will be done. However, functions are being delivered with JES2, such as the Spool encryption and compression, JES2 policies, Disk Reader capability, and enhancements on JES3 JECL support on JES2.

- Do I need to (or should I) migrate from JES3 to JES2?

  The migration should be a business decision that is based on the functions that your organization uses. Migration to JES2 is not currently required. However, remember that the z/OS release after z/OS 2.4 will be the last z/OS release that will include JES3.

- If I am new to z/OS, which JES should I use?

  Customers are highly encouraged to use JES2 because JES3 will not be available in the near future.

- I was considering migrating from JES2 to JES3. Should I?

  IBM recommends that you avoid migrating from JES2 to JES3.

1.2 Job Entry Subsystem

Job Entry Subsystem (JES) is a required and strategic part of the z/OS operating system. Up to the z/OS release after z/OS 2.4, IBM offers two JES choices: JES2 and JES3. In the past, JES3 was considered to be the premium choice and continues to have license fees today.

In the early days of JES2 and JES3, the differences between the two were more obvious than they are today. JES3 was originally developed to assist installations that needed to manage multiple IBM MVS™ images. However, JES2 is used by most z/OS customers and became nearly a superset of JES3 functionality.

Today, JES3 functions, such as multi-system consoles, automatic tape sharing, dynamic initiators, and workload balancing, can be provided by the operating system. Therefore, they are available to installations that run JES2. This difference has left some JES3 installations wondering whether the premium they pay in licensing fees to run JES3 is still worthwhile.

Also, with the challenges for the use of integrated technologies and digital transformation, JES2 and JES3 must provide functionality, such as high availability, fault toleration, or digital integration capabilities with new environments.

IBM stated that JES2 is the strategic JES for z/OS and any development of new functions in spooling subsystems occur primarily in JES2. JES2 supports unique features in the following areas:

- Availability: Spool migration and online merging of spool volumes
- Function: Support for email notification when a job completes
- Security: SPOOL data encryption and compression

1.3 JES2 availability

One important capability that is required by JES is to continue processing, even if one or more sysplex member fails with no effect on the total environment.

For this reason, when JES2 is configured as a Multi Access Spool (MAS, also called JESPlex), all members can monitor and continue processing after one sysplex member fails. A JES2 MAS and a JES3 JESPLEX must be entirely contained within one sysplex.

Since z/OS 2.3, the measurements and monitoring of JES2 key resources can be done and made available to the user by using commands or reporting mechanisms. This feature can help in reducing or eliminating resource exhaustion in JES2.

Reserved space can be set aside for use in recovering the environment when resources are nearing exhaustion. Thresholds can be set and alerts issued well before resources are exhausted and a possible outage occurs.

1.3.1 MAS members

In a JES2 MAS, all members are peers and can perform all JES2 functions. You can choose to run certain functions on one or more members; for example, network job entry (NJE) or printing.

Any member can join or leave an MAS at any time without affecting other members. Every MAS is defined as having 32 members, even if only one member exists. Members do not have to be predefined before starting. A new member can define itself to a MAS as part of its initialization.

One implication of this processing is that a single system environment does not exist in JES2. Even when only one member is active, it is considered a single-member MAS. As such, JES2 processing is the same if one or multiple members are in a MAS.

JES2 and JES3 use a single main task to do most of the work that must be done in the JES address space. In JES2, each member's main task does the work that is needed by that member. In JES3, one main task on the global must do the work that is needed by all members of the JESPlex. This approach can become a bottleneck for processing.

JES2 uses the z/OS cross-system coupling (XCF) services for communicating JES2 member status and other data among the JES2 XCF group members in a MAS configuration. Each member of a JES2 MAS starts independently, joining the JESXCF XCF group and uses group and member information from the initialization deck. This policy exists to inform any other MAS members of its existence and to open a communication path to other members. All members of a MAS must be contained within the same sysplex.

Members in the JESXCF group can communicate with each other by using z/OS XCF services. For example, this communication allows a members of the group to obtain access to the checkpoint data set lock and read the checkpoint into memory and process it.

1.3.2 JES2 resiliency

The resiliency features in JES2 are important for high-availability environments.

Checkpoint data set

The JES2 checkpoint holds the primary job and output queues, data that is needed to manage the spool, and other areas that JES2 requires to keep members synchronized. It also contains data that is needed to start or restart a member. The JES2 checkpoint process performs the following functions:

- Job and output queue back up to ensure ease of JES2 restart
- MAS member-to-member workload communication to ensure efficient independent JES2 operation

Checkpointing is the periodic copying of a member's in-storage job and output queues to the checkpoint data set, which can be on a DASD volume or a coupling facility structure. Checkpointing ensures that information about a member's in-storage job and output queues is not lost if the member loses access to these queues. Loss of access might result from hardware or software errors. Because all members in a JES2 MAS configuration operate in a loosely coupled manner, each member can select work from the same job and output queues.

In a MAS environment, the checkpoint data set backs up the job and output queues and links all members. It is the commonly accessible repository of member activity that allows each member to communicate and be aware of the current workload.

The checkpoint data set contains a record of member values that describe the overall configuration of the MAS environment and specific characteristics and information that describes the status of each member. The checkpoint allows all members to access and update (write to) the checkpoint data set. It also allows all members to refresh their in-storage queues by reading from the checkpoint data set.

Because checkpoint is the JES2 component that contains the major JES2 data areas, it requires short access times and specific capabilities for automatic recovery in case of failure.

Checkpoint reconfiguration

The checkpoint reconfiguration facility allows the dynamic redefinition of the checkpoint data set definitions for the JES2 multi-access spool (MAS) configuration. The reconfiguration allows the installation to perform the following tasks:

- Replace a checkpoint data set.
- Discontinue the use of a checkpoint data set.
- Resume the use of a previously suspended checkpoint data set.
- Start the use of a new checkpoint data set.

It also ensures that no data is lost if a checkpoint error occurs. Data is written from data in memory and from the member that has the most recent CKPT, which ensures currency. For CKPT on CF, you can use the XCF **`rebuild`** function to move from one CF to another. With JES2 **`reconfiguration`**, you can move from CF to DASD or vice versa.

Also, the JES2 can automatically start a reformat reconfiguration function that is internally triggered if a CKPT data error is detected. It reformats the CKPT with current checkpoint data and does not change what data sets (structures) are being used.

The use of the coupling facility (CF) to hold the checkpoint results in the following benefits:

- Better performance and lock management
- Ability to use z/OS functions to move structures from one CF to other
- JES2 usage that is consistent with other CF users

Health Check for JES2 checkpoint reconfiguration

The z/OS 2.4 brings a new health check to identify issues with checkpoint configuration that prevents JES2 from automatically recovering from device errors. Reports that are generated by the check can be viewed along with exception messages that are issued to the system console. They also are used to identify what actions can be performed to improve the checkpoint settings to avoid potential outages.

The check runs by intervals and can detect issues that might arise over time, such as the amount of space that is available on the backup checkpoint volume becoming insufficient to hold a backup checkpoint. It is refreshed automatically when checkpoint settings are changed by a **`$T CKPTDEF`** operator's command or by the checkpoint reconfiguration dialogue.

The following checks are run by Health Check:

- Are all checkpoints defined (CKPT1, CKPT2, NEWCKPT1, and NEWCKPT2)?
- Are both checkpoints currently in use (CKPT1 and CKPT2)?
- If backup checkpoints are defined and exist on DASD, are they large enough to hold the current checkpoint?
- If backup checkpoints are defined but do not exist on DASD, does enough space exist on the volume to create them?
- If running in DUPLEX checkpoint mode, has at least one member specified DUPLEX=YES?
- Is OPVERIFY=NO specified to allow JES2 to handle device errors automatically without operator intervention?

SPOOL privileged space

The SPOOL is the bulk data repository in JES2. It primarily contains JES-managed data sets that include job input (in-stream data) and output (SYSOUT). This area of functionality includes job-oriented data sets, such as the JCL, the output of the converter (internal text), and restart information (the job journal). It also contains several job-related control blocks that are used to manage the characteristics of a job and the data sets that it owns.

Precisely because JES2 is an essential component for all z/OS processing, one of the most unwanted situations is for the spool space to be exhausted. This condition affects all systems.

In z/OS 2.3, JES2 spool management can reserve spool space to deal with resource exhaustion. Approximately 1% of spool space is reserved by JES2 for the following items:

- Spool resources
- Jobs queue elements (JQEs)
- Job output elements (JOEs)
- Block extension reuse tables (BERTs)

This “privileged space” is sufficient to allow a user to perform the following functions:

- Log on to the system
- Submit jobs
- Resolve causes of exhaustion

Typically, privileged jobs, STCs, and TSO logons use this privileged space. Therefore, the normal management environment is still available during a resource shortage. To deliver this benefit, JES2 uses an “emergency subsystem” that is another portal into the main subsystem.

With this emergency subsystem, a privileged TSO user ID can be logged on through SUBSYS(xxxx) by using the `TSO LOGON` command. This user account removes the cause of exhaustion, with no outage during the return to normal processing.

Initialization data set checker

Whenever a JES2 initialization deck must be changed, the question of whether it will work on next IPL is raised and requires actions to double check all of the coded specifications.

The initialization data set checker allows installations to verify their initialization data sets without having to start a JES2 subsystem. The process can detect syntax errors in initialization statements and problems with settings that might prevent JES2 from starting.

This initialization data set checker can avoid outages that are caused by JES2 initialization failure because of initialization parameters errors. The initialization data set checker analyzes to see whether current specifications are reasonable and no errors are found.

The checker can be used in the following ways:

- By specifying CHECK start PARM value (for example, PARM=‘cold,check’)
- By using the alternative entry point HASJESCK (for example, PGM=HASJESCK)

The initialization data set checker is useful to test some initialization exits that run during JES2 initialization. The initialization data set checker loads all installation modules that are specified by using a LOAD(module-name) initialization statement.

This approach allows exits to define and process any installation- or vendor-defined initialization statements. Because the checker is not starting a JES2 subsystems, all modules are loaded in private storage (even if the LOAD statement specifies common storage). The normal JES2 initialization exits (0, 19, and 24) are called to perform any validation processing that might be needed.

1.4 JES2 job management

JES2 includes a function to make it possible to write JCL for jobs that run in a specific order without the need for an external job scheduling package. This function is called *job execution control* (JEC). Although not intended to replace job scheduler, this function simplifies breaking down large, complex multistep jobs into multiple jobs that can eventually be placed under the control of a job scheduler.

It is also intended to ease applications that can analyze JCL while it is being submitted and break down the steps into separate, dependent jobs. This function helps users that are running JES3 and JES2 by providing similar functions as the JES3 dependent job control (DJC) in the JES2 environment.

The principal entity that controls job execution within JEC is a *job group*. A job group is defined by the JOBGROUP JCL statement.

The following JCL statements provide JEC support:

- JOBGROUP: Creates a job group.
- ENDGROUP: Denotes the end of the job group.
- GJOB: Defines a job within a job group.
- JOBSET: Provides a convenient method to define and reference a set of jobs with identical dependencies.
- SJOB: Defines a single job within the job set.
- ENDSET: Denotes the end of the job set.
- BEFORE: The current job must run before the jobs or job sets that are listed with this JCL statement.
- AFTER: The current job must run after the jobs or job sets that are listed with this JCL statement.
- CONCURRENT: Defines a set of jobs or job sets that must run at simultaneously on the same JES2 MAS member.
- SCHEDULE: Associates a job with a job group.

The group of keywords BEFORE=/AFTER=/DELAY= in the SCHEDULE JCL statement facilitates the ad hoc sequencing of jobs without accessing a static JOBGROUP, which is also known as *dynamic job sequencing*. Use this option to create dynamic scheduling on batch work flows when you process a job by using one of the following methods:

- By normal submit command
- By job scheduling software, such as IBM Tivoli® Workload Scheduler

In some job scheduling software, these keywords can be set dynamically by the software when the specified internal processing conditions are met. This software uses its own job editor processing to change the sequence or name of scheduled jobs into a JOBGROUP.

By using this context, production teams can create complex job scheduling, such as calendar dates, time spans, and conditions for job completion.

It is also possible to define how JES2 Input Processor handles JES3 JECL statements. If you allow JES2 to process JES3 JECL, the conversion of JES3 job streams that use Dependent Job Control (DJC) can be minimized or even avoided. JES2 Input Processing creates a JOBGROUP and adds all jobs in the same NETID to this group.

1.4.1 Email Delivery Service

JES2 Email Delivery Services (EDS) is a JES2 function that accepts email messages from JES2 interfaces and delivers them to the intended recipients (the email addresses). After a job ends, if conditions that are specified by the WHEN keyword of a NOTIFY JCL statement are satisfied and the notification method is set as email, JES2 sends job end message by an email message; for example, notify the security administrator if job security validation fails.

JES2 provides the following interfaces for sending email messages:

- The NOTIFY JCL statement specifies conditions and delivery method for job termination notification. One of the supported delivery methods is the email message process after a job ends.
- The Notify user message service (SSI 75) allows an application to send a message to a user. One of supported delivery methods for the message is email.

The following stages are used by JES2 EDS to processes email messages:

1. The email messages are stored on JES2 SPOOL when email messages are accepted for delivery.
2. Email messages are read from JES2 SPOOL and are delivered to the intended destination.

Separating email processing by using these stages allows JES2 to accept email messages, even if the environment does not allow immediate delivery of the email; for example, TCP/IP services are not available or the email server is not accessible. In addition, this separation helps to protect accepted email messages from system failure.

Most of JES2 EDS processing is performed in a separate address space. The name of the address space uses the format <subsystem>EDS, where <subsystem> is a subsystem name that is used by JES2. For example, if the subsystem name is JESA, the address space name is JESAEDS.

The following main functions are provided by JES2 EDS address space:

- Accepts email messages and saves on JES2 SPOOL.
- Reads email messages from JES2 SPOOL and sends them to intended recipient.

JES2 EDS accepts email messages and stores them in the JES2 SPOOL on any JES2 MAS member. No extra z/OS functions are required for that stage of email processing.

To deliver email messages, JES2 EDS relies on the services that are provided by z/OSMF. The z/OSMF server does not have to be active on the same SYSPLEX member. All MAS members can access the same z/OSMF server active anywhere in the SYSPLEX if communication to the z/OSMF server is possible.

1.5 JES2 security

Security in a data processing environment involves controlling and auditing access to resources that are important to your installation. In the JES2 environment, these resources include the following examples:

- JES2-owned data sets
- Input (from nodes, remote workstations, readers, offload devices, and commands)
- Job names
- System input/output that is on spool (SYSIN/SYSOUT)
- Output (to nodes, printers, punches, remote workstations, and offload devices)

JES2 provides a basic level of security for some of these resources through initialization statements. For example, each node in a network can be defined as having a certain level of control over work at each of the other nodes in the system. This level of control can give one operator limited control over each of the other nodes.

The control that is available through initialization statements can be broadened by implementing several JES2 exits that are available for this purpose. You can implement a more comprehensive security policy by using the System Authorization Facility (SAF) component of z/OS and a security product, such as IBM Resource Access Control Facility (RACF®). SAF provides a link to the security product to define any other security controls that your installation might require.

JES2 passes information to SAF to perform password validation, request authority to access a resource, and determine security information in various environments. When SAF and the security product indicate a decision on a security request, JES2 bypasses its own security processing.

1.5.1 Data encryption

The need for creating secure archived copies of business data is a critical security concern. Encrypting data that can be recovered at any time offers a high degree of privacy protection from unwanted access. For this reason, the use of data encryption became the most important feature to be used for data security.

NJE encryption

Because the NJE connection is available over TCP/IP, most installations use this option without any security mechanism as SSL or AT-TLS to encrypt the data that is transferred between nodes.

SSL and TLS provide excellent security, from a TCP/IP standpoint. These protocols encrypt data on unsecure links and ensure that the peer node at the other end of the connection is who it claims to be from a TCP/IP standpoint.

However, from an NJE standpoint, you might need more security to ensure that the peer node is who it claims to be. You can specify NODE and LINE parameters for connections, but these passwords are exchanged in clear text in sign-on records. They might be compromised if they are sent into an unsecure network.

The SECURE=REQUIRED option on NETSERV statement indicates that NETSERV should accept only connection requests with a secure protocol in use, such as TLS/SSL.

With this option, the NJE traffic over TCP/IP connections can be secure and the data is protected, including the use of a virtual private network between the NJE nodes.

The JES2 can encrypt a particular line (in which case everything that is sent on that line is encrypted) or a particular transmission. End-to-end encryption is the process of encrypting a telecommunication line. When you use TCP/IP for performing NJE, you can define a policy agent for the network and exchange digital certificates between nodes that are in network.

SPOOL encryption

Starting with z/OS 2.4, JES2 includes a function to encrypt and compress spool data, which improves security on sensitive information and reduces the storage requirements. The compression also can improve the performance of managing the data.

The spool data is encrypted by using the z Systems hardware encryption through existing policy management (without application changes to the programs) by defining a record in the CKDS data set that can be identified and accessed by using a 64 character key label.

The JCL parameter DSKEYLBL or JESJOBS class profiles can be used to identify selective SYSOUT and in-stream data sets to be encrypted. Data to be encrypted first is compressed, which provides storage efficiency.

To select spool data to be compressed, the new COMPRESS= option on OUTCLASS(*x*) statement is used even, if no data encryption is required.

1.5.2 Passphrase support

The requirement for z/OS to accept the use of passphrases for user ID authentication added another security level and the possibility of integration between z/OS and other platforms.

The use of passphrase for user authentication on a JCL JOB card is accepted by JES2 as an extra security mechanism and compatibility between platforms. The keyword PASSWORD= on job card supports pass phrase. Consider the following points:

- If the data in the password field is 1 - 8 characters, it is considered as a password.
- If the data in the password field is 9 - 100 characters, it is considered as a passphrase.

1.6 JES2 and JES3 compatibility

For a system that has only a single z/OS image, JES2 and JES3 perform similar functions: they read jobs into the system, convert jobs to internal machine-readable form, select jobs for processing, process output, and purge jobs from the system.

However, for a system that has multiple processors, noticeable differences exist in how JES2 exercises independent control over its job processing functions. Each JES2 processor controls its own job input, job scheduling, and job output processing. In contrast, JES3 exercises centralized control over its processing functions through a single global JES3 processor.

SMF84 record

Record type 84 contains information that is collected by JES2 or JES3 monitors. This record is intended to provide insights into what the subsystems are doing during the interval the record represents.

In JES3, the information is collected by the JES3 monitoring facility (JMF). When JMF is called with the SMF option selected, these records are generated for each JMF interval. SMF records can be produced on the global and local processors.

In JES2, the information is collected by the JES2 health monitor. The records are generated by each JES2 subsystem address space at the top of every hour.

The SMF record (subtype 21) can be used for real-time management of JES2 memory resource usage. The record provides information for the following areas:

- The memory usage contains information about each memory section in the JES2 address space.
- The resource usage contains information about various resources JES2 manages.

JES3 JECL support

JES2 supports the processing of JES3 JECL statements with native support or translation into supported statements (see Table 1-1).

Table 1-1 JES3 JECL statements that are supported on JES2

JECL statement	Supported level
//*DATASET	Tolerated, but not supported.
//*ENDDATASET	Required if //*DATASET used.
//*FORMAT	Partially supported (converted to OUTPUT JCL card).

JECL statement	Supported level
//*MAIN	Partially supported (supported in z/OS 2.2).
//*NET	Partially supported (converted to JES2 job group).
//*NETACCT	Fully supported.
//*OPERATOR	Supported, but message text ends in 71, not 80.
//**PAUSE	Not supported, ignored if present.
//*PROCESS	Tolerated, but not supported.
//*ENDPROCESS	Tolerated, but not supported.
//*ROUTE XEQ	Fully supported.

The activation of this support can be done at two levels: first, it gives the ability to handle the JES3 JECL syntax as JECL and not a comment; second, it gives control over how each JECL statement is processed.

JES2 includes support for the primary (//*) and alternative (/*) prefix for JES3 JECL, except to //*ROUTE XEQ that does *not* support the alternative /*ROUTE XEQ syntax because of conflicts with the JES2 /*ROUTE XEQ statement.

2

Terminology differences

This chapter describes some of the terminology you are likely to encounter in the rest of this IBM Redbooks publication. We focus in particular on cases where the same term is used in JES2 and JES3, but with different meanings.

This chapter includes the following topics:

2.1 JES3 terminology

As a JES3 user, you are familiar with JES3 terms, such as *processor* (global processor, local processor). This term refers to the hardware instance (partition) that contains software to interpret and process instructions.

2.2 Different use of terms

JES2 and JES3 have long and somewhat independent histories. In some situations, the use of terminology might cause some confusion when you move from one JES to the other. In some cases, the same term is used to mean different things. In other cases, two different terms might be used to describe the same thing.

This section brings all of these cases together to help avoid confusion as you read the remainder of this book.

2.2.1 Non-specific JES2 and JES3 references

Often it is useful to refer to JES2 and JES3 in ways that are non-specific, as shown in the following terms:

- JES

  This term is a non-specific reference to a JES. It is used to refer to concepts that apply to both JESs. For example, “Each z/OS image must have a JES subsystem to process jobs.”

- JES-neutral

  Some functions work the same in JES2 and JES3. These functions are often referred to as *JES neutral*. For example, security processing is performed by the security product in a way that is JES-neutral.

- JES-agnostic

  Much like JES neutral, this term applies to something that uses JES2 service in a way that is unaffected by the type of JES that you are running. For example, “One of the goals in preparing for a migration to JES2 is to make your JCL JES-agnostic.”

2.2.2 Collections of JESes

JES2 and JES3 feature the concept of a collection of JES address spaces that share a single work queue. However, the following terms that are used to refer to the collection varies with the JES that is referenced:

- MAS

  A Multi-Access SPOOL, this term is used by JES2 to refer to the collection of up to 32 JES2 address spaces that share a single JES2 work queue.

- Complex

  JES3 uses this term to refer to its collection of a single JES3 Global and up to 31 Locals that share a single JES3 work queue.

- JESplex

  A JES complex is a JES-neutral term that refers to a JES2 MAS or a JES3 complex. You can also modify this term with the type of JES, such as a JES2 JESplex.

The term that you use to refer to one of the JES address spaces in the collection also differs by JES, as shown in the following examples:

- Member

  JES2 address spaces in a MAS are referred to as *members of the MAS*. This term is also used to refer to members of a JESplex.

  In JES2, the member name defaults to the SMF ID of that image. You can override that name with a name of your choice, but the name is limited to four characters.

  You can have multiple JES2 instances in a single MVS image (poly-JES). The term poly-JES refers to the concurrent operation of multiple copies of JES2.

  z/OS allows more than one JES2 subsystem to operate at a time if one subsystem is designated as the primary subsystem and others are identified as secondary subsystems. Secondary JES2s can be useful in testing user modifications while the primary JES2 is being used for production.

- System

  JES3 address spaces in a complex are often referred to as systems. This terminology was developed because only one JES3 address space per z/OS image is available.

- Main

  This term is another name for a member of a JES3 complex. This member can be a Local or a Global.

  In JES3, the name of a JES3 Global or Local can be up to eight characters long.

JES2 supports up to three consecutive releases of JES2 that are running in the same MAS. JES2 enforces this rule and ships service to previous releases so that they can coexist with the newer releases.

JES2 also supports a more liberal migration policy. This migration policy covers migrating from one release of JES2 to a new release with a warm start.

2.2.3 JES startup processing

Operationally, JES2 can be started by using one of two methods. You can specify **PARM=COLD** or **PARM=WARM** (the default). A cold start clears all JES2 SPOOL and checkpoint data areas (delete all jobs) and as with a JES3 cold start, is rarely done. Warm starts are the normal way to start JES2. How JES2 processes a warm start depends on the environment JES2 discovers during initialization. Depending on the environment, a JES2 warm start performs different processing.

Regardless of the type of start, JES2 always reads its parameters from the PDS members that are pointed to on the HASPPARM DD statements or the default PARMLIB concatenation when it is starting. When reading from default PARMLIB concatenation, JES2 uses HASjesx as member name, where jesx is the subsystem name that is associated with it. It then compares the information that is found in the parameters with the information it receives from the checkpoint and from the other members of the MAS. If it encounters a parameter that requires a more disruptive type of start, it might issue an HASP442 message, which informs you that the parameter was ignored.

JES2 can perform the following types of starts:

- Cold start

  This start occurs when you specify **PARM=COLD** when JES2 is started. The JES2 spool is cleared of all contents. This type of start requires that all the members of the MAS are stopped.

  > **Note:** On the system that is performing the cold start, you must take one of the following actions:
  >
  > - Perform an IPL.
  > - Completely stop JES2 and then restart it and specify that it performs a cold start.

  Given that all work on the system must be stopped before this start is performed, little difference exists between stopping and restarting JES2 to perform the cold start and IPLing that system.

- Warm start (single system)

  This start occurs when you specify **PARM=WARM** when JES2 is started and the starting JES2 member is joining a MAS with other active members. The JES2 checkpoint is read in and processed. Any work that might be associated with this member from a previous instance is reset (marked as no longer actively being processed).

- Quick start (single system)

  This start occurs when you specify **PARM=WARM** when JES2 is started. This process is the same as a warm start except that no work is associated with this member from a previous instance. This process occurs if the member:

  – Was shut down cleanly by using a **$P JES2** command
  – Is starting after an all-member warm or cold start
  – Had its work reset by using a **$E MEMBER** command or through the **AUTOEMEM** process

- Warm start (MAS-wide)

  This start occurs when you specify **PARM=WARM** when JES2 is started and the starting member is the first (only) active member of the MAS. As with all other warm starts, the checkpoint is read in and processed. If any entry in the work queue indicates it is active, it is reset then. In addition, certain operating parameters can be reset only on this type of start.

- Hot start

  This start occurs when you specify **PARM=WARM** when JES2 is started and a previous instance of the JES2 address space had ABENDed and no intervening IPL occurred. As with all other warm starts, the checkpoint is read in and processed.

  Work in the job queue that is associated with processes that were ended when the JES2 address space was ABENDed are reset. However, work that is associated with active address spaces (running jobs, internal readers, and so on) is not reset. That work continues normal processing.

  > **Note:** When working with a secondary subsystem, all start options are available and affect only the secondary subsystem without any effect on the primary JES2 subsystem.

JES3 uses the following similar terminology, but the effect can be different. For JES3, the start type is set in the response to message IAT3011:

- Cold start

  During a cold start, the initialization deck is read to determine the configuration. The SPOOL is initialized, and any jobs that were in the system are lost. A JES3 cold start requires that every z/OS in the JES3 complex is IPLed.

- Warm start

  A warm start also requires an MVS IPL before it is allowed. The configuration is determined from the initialization stream. Most job processing resumes after this less intrusive restart. As with a cold start, a JES3 warm start also requires that every z/OS in the JES3 complex is IPLed.

- Hot start

  During a hot start, the initialization stream is *not* read. Instead, the configuration information is read from control blocks that are stored in the spool. A JES3 hot start does not require that you IPL z/OS. If a system is IPLed and then JES3 hot starts, job processing resumes. Job processing can continue across a JES3 hot start if the system is not IPLed.

- Hot start with refresh

  Hot start with refresh is similar to a hot start except that the initialization stream *is* reread. This process allows for initialization deck statements to be altered without requiring a warm or cold start and the associated complex-wide IPL.

- Restart with analysis

  Hot and warm JES3 restarts allow for an extra analysis option to be specified. When analysis is requested, more verification is performed for jobs on the job queue and invalid jobs can be purged.

- Warm start with replace

  This start performs the same function as a warm start. In addition, it allows you to replace a spool data set.

2.2.4 JES parameter statements

Both JESes include the following parameter statements that define objects to JES and set operating parameters:

- Init deck

  JES2 initialization data set. This parameter is read in on every start of a JES2 address space. The format of statements in the JES2 initialization data set deck is the same as the corresponding operator command and the display command for the statement.

- Inish deck

  JES3 initialization stream. This parameter is read when JES3 first initializes on a system and on a hot start with refresh.

2.2.5 SYSOUT processors

Both JESes support sending SYSOUT from SPOOL to physical or logical devices for processing. The following processors are referenced:

- Printer

  In JES2, a printer (JES-controlled or FSS-managed) is referred to as a *printer*. Applications that use SAPI are referred to as SAPI applications or SAPI threads. SYSOUT is commonly referred to as being placed on the print queue or the ready queue.

- Writer

  In JES3, a printer (JES-controlled or FSS-managed) or an application that uses the *SYSOUT API* (SAPI) is referred to as a *writer*. SYSOUT is commonly referred to as being placed on the writer queue.

2.2.6 Remote workstations

A remote work station that connects to JES is referred to as:

- RJE: Remote Job Entry (RJE) in JES2
- RJP: Remote Job Processing (RJP) in JES3

2.2.7 JES threads

Both JESes include a main task that is shared by multiple threads or processes. Externals also control the number of particular types of the following threads in both JESes:

- PCE

  JES2 processor control element. This control block represents a thread or process that is running under the JES2 main task. Each PCE performs a function (such as execution services), processes a job phase (such as the purge phase), or manages a device (such as a printer).

  Some PCEs are created at initialization based on keywords on **PCEDEF** or other internal constants. Others can be created with commands (such as **$ADD PRINTER**). Exit code or installation load modules can also define and create PCEs as part of their processing.

- DSP

  JES3 dynamic support program. A DSP represents code that performs a small piece of job processing. Most JES3 job processing is performed by IBM written DSPs. These units of work, along with FCT entries, provide the basis for JES3 subsystem multitasking.

- FCT

  JES3 function control table. The JES3 main task processing scans a priority-ordered chain of FCT entries, looking for any FCT entries that represent active work to be done. Each FCT entry points to a DSP that is called to perform the work. Because FCT entries reference DSPs, the two terms are sometimes used interchangeably. Multiple FCTs that reference the same DSP can be inserted into the DSP chain. Some FCT entries are permanently stored in the FCT chain, and some are added and removed only as needed.

2.2.8 Multiple JES2 instances

JES2 supports running multiple instances of JES2 on a single z/OS image. The extra instances can be in a separate MAS to the primary JES2, or they can be in the same MAS as the primary JES2. You might want to have more than one JES2 instance for the following reasons:

- To test new functions on a production system, but separate from the production MAS.
- To offload functions, such as NJE or printing from the primary member to a secondary.
- To provide easy access to a secondary MAS on a production image.

The following terms are used to describe this concept:

- Primary JES

  This term refers to the JES that is the primary subsystem on a particular z/OS image. Only one primary JES is available.

  JES2 can run as a primary or secondary JES. JES3 can *only* run as a primary.

- Secondary JES

  This term refers to a JES2 subsystem that is not the primary JES subsystem. Many secondary subsystems can be on a z/OS image. JES2 subsystems always are available because JES3 does not support running as a secondary subsystem.

- PolyJES

  This term refers to the process of running multiple JES instances on a single z/OS image.

- Alternate JES

  This term is another name for a secondary JES2 subsystem. Alternate is often used when more than two JES instances are used on a single z/OS image. For example, “This system has three JES2 address spaces; one primary and two alternates.”

2.2.9 JES initialization statements

JES2 and JES3 fundamentally provide the same function: batch job handling. They both depend on a set of initialization statements to define the configuration to them. Therefore, it is not surprising that JES2 and JES3 include initialization statements that are similar. In some cases, they use identical keywords. At times, these keywords have the same meaning; other times, they have subtly different meanings.

3

Convergence of JES2 and JES3

This chapter provides information about functions and features that are available in JES3 that are provided by JES2. Also described is how this convergence processing is being adopted to make both products more compatible and transparent from a user viewpoint.

The chapter also covers the functionalities that are different between JES2 and JES3 and how they can be addressed when converting from JES3 to JES2.

This chapter includes the following topics:

- 3.1, “JES2 to JES3 compatibility” on page 24
- 3.2, “JES2-only functions” on page 31
- 3.3, “JES3-only functions” on page 34

3.1 JES2 to JES3 compatibility

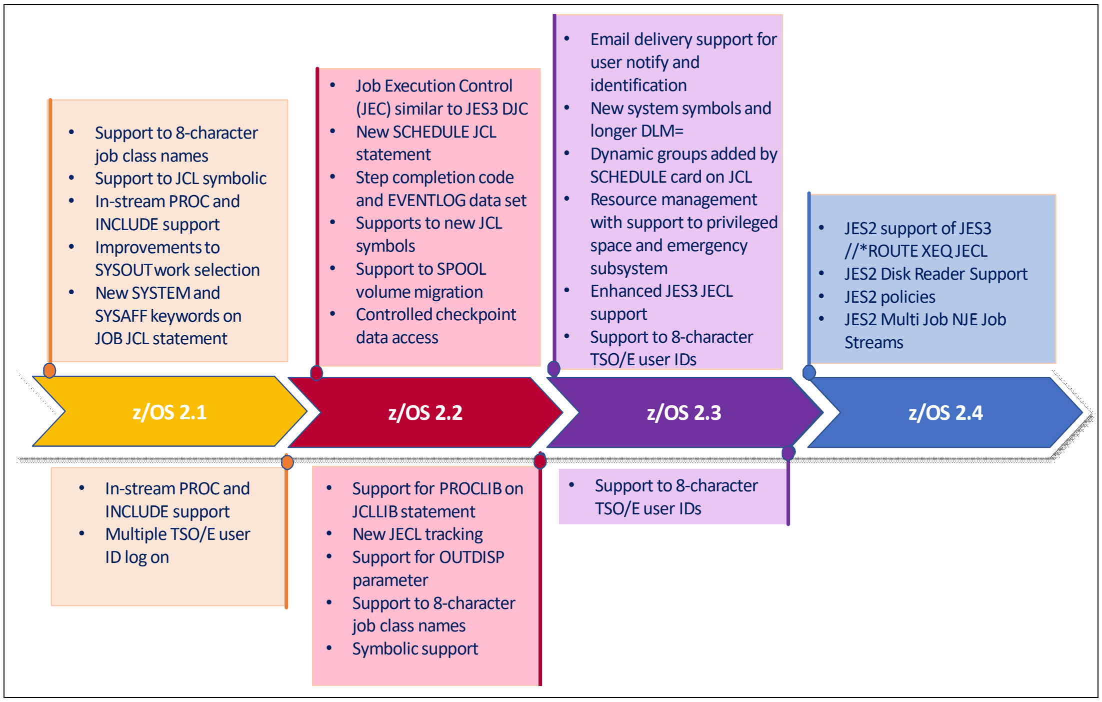
Over time, the differences between JES2 and JES3 decreased. Although they have different processing mechanisms, they can perform the same actions from a user's view point.

In the last releases of z/OS, you can see that these differences are becoming minimal, as shown in the following examples:

- JES2 now includes Job Execution Control (JEC), which performs functions that previously were available in JES3 DJC only.
- The use of SCHEDULE evolved.
- JES3 JECL support is enhanced in every new JES2 release.
- Disk reader and multi job NJE job stream support was added in z/OS 2.4.

Also, some JES3 functions, such as multi-system consoles, automatic tape sharing, dynamic initiators, and workload balancing, can be provided by the operating system and are available to JES2 installations.

The convergence between JES2 and JES3 is increasing and both products are performing similar functions (see Figure 3-1). The enhancements to JES2 that were introduced in each new release are making it easier to migrate from JES3 to JES2.



- Support to 8-character job class names
- Support to JCL symbolic
- In-stream PROC and INCLUDE support
- Improvements to SYSOUT work selection
- New SYSTEM and SYSAFF keywords on JOB JCL statement

- Job Execution Control (JEC) similar to JES3 DJC
- New SCHEDULE JCL statement
- Step completion code and EVENTLOG data set
- Supports to new JCL symbols
- Support to SPOOL volume migration
- Controlled checkpoint data access

- Email delivery support for user notify and identification
- New system symbols and longer DLM=
- Dynamic groups added by SCHEDULE card on JCL
- Resource management with support to privileged space and emergency subsystem
- Enhanced JES3 JECL support
- Support to 8-character TSO/E user IDs

- JES2 support of JES3 //*ROUTE XEQ JECL
- JES2 Disk Reader Support
- JES2 policies
- JES2 Multi Job NJE Job Streams

z/OS 2.1

z/OS 2.2

z/OS 2.3

z/OS 2.4

- In-stream PROC and INCLUDE support
- Multiple TSO/E user ID log on

- Support for PROCLIB on JCLLIB statement
- New JECL tracking
- Support for OUTDISP parameter
- Support to 8-character job class names
- Symbolic support

- Support to 8-character TSO/E user IDs

Figure 3-1 Convergence between JES2 and JES3

3.1.1 JES2 health monitor

The JES2 health monitor is a service that allows for the monitoring of the subsystem status. By monitoring the status of the subsystem, JES2 can identify situations when the subsystem is not behaving the way it is expected to behave. JES2 uses the data that is provided by the monitor to notify the operator to determine the causes of the problems within the susbsystem.

However, remember that the monitor is not a performance monitor; it can help you identify possible causes that affect the subsystem performance.

The health monitor automatically starts when JES2 is initialized and ends when JES2 ends.

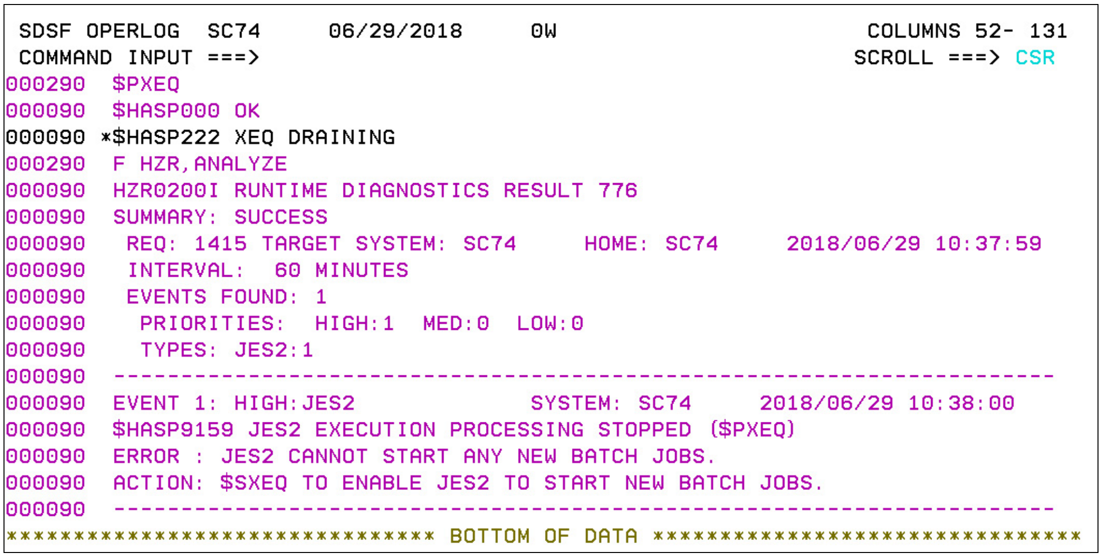
The health monitor samples JES2 processing to collect data and provide information when JES2 is not responding to commands or the problems are not easy to diagnose. Such situations can be basic, as shown in the following examples:

- A command that is taking an unexpected amount of time to complete.
- A legitimate “bug” in JES2.
- An exit routine problem.
- Problems with checkpoint.

Other code running in the JES2 address space

z/OS Runtime Diagnostics provides JES2 diagnostic information to assist the system programmer in identifying symptoms that contribute to “sick, but not dead” behavior.

JES2-detected health exceptions are added as events in Runtime Diagnostics, which is started by using the **F HZR,ANALYZE** system command (see Figure 3-2).



```
 SDSF OPERLOG  SC74     06/29/2018     0W                        COLUMNS 52- 131
 COMMAND INPUT ===>                                             SCROLL ===> CSR
000290  $PXEQ
000090  $HASP000 OK
000090 *$HASP222 XEQ DRAINING
000290  F HZR,ANALYZE
000090  HZR0200I RUNTIME DIAGNOSTICS RESULT 776
000090  SUMMARY: SUCCESS
000090   REQ: 1415 TARGET SYSTEM: SC74     HOME: SC74     2018/06/29 10:37:59
000090   INTERVAL:  60 MINUTES
000090   EVENTS FOUND: 1
000090    PRIORITIES:  HIGH:1  MED:0  LOW:0
000090    TYPES: JES2:1
000090  ----------------------------------------------------------------------
000090  EVENT 1: HIGH:JES2            SYSTEM: SC74     2018/06/29 10:38:00
000090  $HASP9159 JES2 EXECUTION PROCESSING STOPPED ($PXEQ)
000090  ERROR : JES2 CANNOT START ANY NEW BATCH JOBS.
000090  ACTION: $SXEQ TO ENABLE JES2 TO START NEW BATCH JOBS.
000090  ----------------------------------------------------------------------
******************************** BOTTOM OF DATA ********************************
```

Figure 3-2   Response of F HZR,ANALYZE command with JES2 event found

Information is gathered about the JES2 subsystem from the JES2 subsystem interface (SSI). Runtime Diagnostics analyzes the information that is received, determines a possible corrective action, and presents this action to the caller on the system console, the hardcopy log, and optionally, to a sequential data set.

Also, when Predictive Failure Analysis (PFA) detects a potential rate that is too low (for the PFA checks that support “too low” processing) and starts Runtime Diagnostics to determine whether events exist, JES2 Health Exception events are returned by PFA when they exist and causes PFA to issue an exception.

SMF record type 84 contains information that is collected by JES2 or JES3 monitors. In JES2, the information is collected by the JES2 health monitor. The records are generated by each JES2 subsystem address space at the top of every hour. The SMF record 84 subtype 21 tracks the resource usage by JES2, which is similar to existing subtype 4. The subtype 4 is the control block utilization section for JES3.

The JES3 Monitoring Facility (JMF) provides several reports that can be used by the system programmers or software support if any specific performance or tuning concerns exist in JES3. The JES3 monitoring facility collects data from the system to see how the installation uses its resources. This information can help detect many performance problems and help you to tune the installation.

A JES3 command **`*X JMF`** starts the facility that can produce a hardcopy report or SMF records.

3.1.2 JES initialization deck checker

In JES2, syntax checking on the initialization parameters data set can be performed by using the initialization deck syntax checker. This JES2 checker can run in batch, as a started task, or linked with no APF authorization requirements. The user under which it runs must read the initialization decks.

The JES2 initialization deck checker can be used in the following ways:

- CHECK start PARM value (for example, PARM='warm,check')
- Alternate entry point HASJESCK (for example, PGM=HASJESCK)

Also, JES2 tends to be more forgiving of syntax errors in the JES2 initialization statements, which provides the operator with an option to resolve many errors during initialization.

The JES3 initialization deck checker is used to validate the format of the JES3 initialization statements. This process is more of an issue in JES3 than in JES2 because of the complexity of the JES3 initialization deck, especially if MDS is used. The IATUTIS program takes input from the IODF and uses that input to validate the syntax of the initialization deck.

Note: This initialization deck checker is enabled by using a JCL to run the program IATUTIS.

For more information about how to use initialization deck checker, including the JCL that is used to run it, see 5.2.1, "Verifying the JES initialization deck" on page 95.

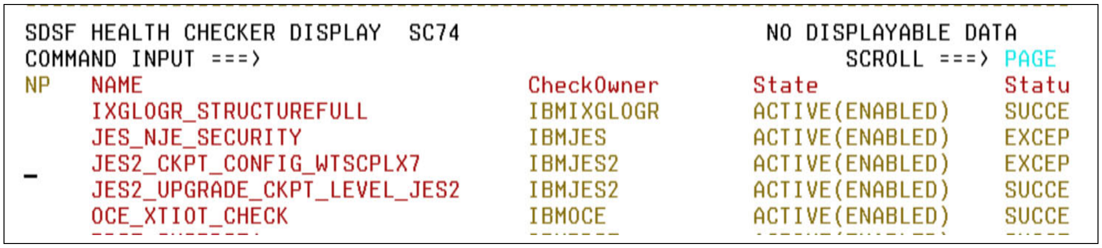
3.1.3 JES2 Health Check for Checkpoint Reconfiguration

Introduced in z/OS 2.4, JES2 Health Check for Checkpoint Reconfiguration is designed to identify problems with the checkpoint configuration that prevent JES2 from recovering from device errors automatically. The check generates reports and messages that are issued to the console that help to take the required actions to improve checkpoint settings to avoid potential outages.

The checker is can detect issues that might arise over time, such as the amount of space on the backup checkpoint volume is not enough to hold a backup checkpoint because it runs on intervals. The check is refreshed if the checkpoint settings are modified.

As part of the initialization processing, JES2 automatically adds the checker to the Health Check.

The checkpoint checker display is shown in Figure 3-3. The current status of the checker is the exception.

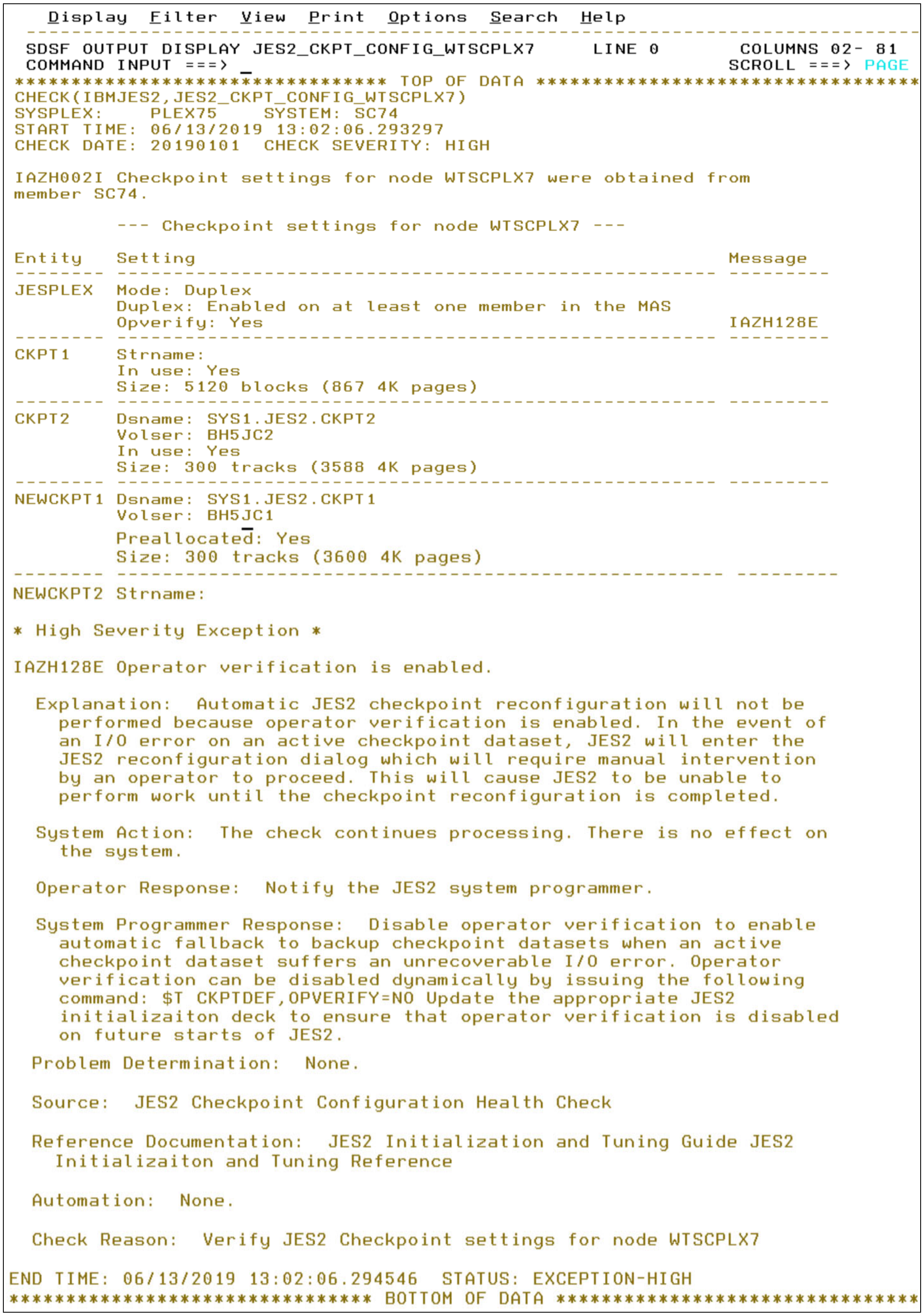


```
SDSF HEALTH CHECKER DISPLAY  SC74                    NO DISPLAYABLE DATA
COMMAND INPUT ===>                                           SCROLL ===> PAGE
NP   NAME                              CheckOwner       State               Statu
     IXGLOGR_STRUCTUREFULL             IBMIXGLOGR       ACTIVE(ENABLED)     SUCCE
     JES_NJE_SECURITY                  IBMJES           ACTIVE(ENABLED)     EXCEP
_    JES2_CKPT_CONFIG_WTSCPLX7         IBMJES2          ACTIVE(ENABLED)     EXCEP
     JES2_UPGRADE_CKPT_LEVEL_JES2      IBMJES2          ACTIVE(ENABLED)     SUCCE
     OCE_XTIOT_CHECK                   IBMOCE           ACTIVE(ENABLED)     SUCCE
```

Figure 3-3 SDSF Health Checker Display

Figure 3-4 on page 28 shows the information that is provided by the checker highlighting the exceptions detected; an operator intervention was requested.

If you review the message, you see that if you maintain the current options, the JES2 automatic checkpoint recovery does not occur. The reason for this failure is that the operator intervention option was enabled. Therefore, if an I/O error occurs on an active checkpoint data set, an operator intervention is requested.



```
   Display  Filter  View  Print  Options  Search  Help
 -------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY JES2_CKPT_CONFIG_WTSCPLX7     LINE 0       COLUMNS 02- 81
 COMMAND INPUT ===> _                                          SCROLL ===> PAGE
********************************* TOP OF DATA **********************************
CHECK(IBMJES2,JES2_CKPT_CONFIG_WTSCPLX7)
SYSPLEX:    PLEX75    SYSTEM: SC74
START TIME: 06/13/2019 13:02:06.293297
CHECK DATE: 20190101  CHECK SEVERITY: HIGH

IAZH002I Checkpoint settings for node WTSCPLX7 were obtained from
member SC74.

         --- Checkpoint settings for node WTSCPLX7 ---

Entity   Setting                                               Message
-------- ----------------------------------------------------- ---------
JESPLEX  Mode: Duplex
         Duplex: Enabled on at least one member in the MAS
         Opverify: Yes                                         IAZH128E
-------- ----------------------------------------------------- ---------
CKPT1    Strname:
         In use: Yes
         Size: 5120 blocks (867 4K pages)
-------- ----------------------------------------------------- ---------
CKPT2    Dsname: SYS1.JES2.CKPT2
         Volser: BH5JC2
         In use: Yes
         Size: 300 tracks (3588 4K pages)
-------- ----------------------------------------------------- ---------
NEWCKPT1 Dsname: SYS1.JES2.CKPT1
         Volser: BH5JC1
         Preallocated: Yes
         Size: 300 tracks (3600 4K pages)
-------- ----------------------------------------------------- ---------
NEWCKPT2 Strname:

* High Severity Exception *

IAZH128E Operator verification is enabled.

  Explanation:  Automatic JES2 checkpoint reconfiguration will not be
    performed because operator verification is enabled. In the event of
    an I/O error on an active checkpoint dataset, JES2 will enter the
    JES2 reconfiguration dialog which will require manual intervention
    by an operator to proceed. This will cause JES2 to be unable to
    perform work until the checkpoint reconfiguration is completed.

  System Action:  The check continues processing. There is no effect on
    the system.

  Operator Response:  Notify the JES2 system programmer.

  System Programmer Response:  Disable operator verification to enable
    automatic fallback to backup checkpoint datasets when an active
    checkpoint dataset suffers an unrecoverable I/O error. Operator
    verification can be disabled dynamically by issuing the following
    command: $T CKPTDEF,OPVERIFY=NO Update the appropriate JES2
    initializaiton deck to ensure that operator verification is disabled
    on future starts of JES2.
  Problem Determination:  None.

  Source:  JES2 Checkpoint Configuration Health Check

  Reference Documentation:  JES2 Initialization and Tuning Guide JES2
    Initializaiton and Tuning Reference

  Automation:  None.

  Check Reason:  Verify JES2 Checkpoint settings for node WTSCPLX7

END TIME: 06/13/2019 13:02:06.294546  STATUS: EXCEPTION-HIGH
********************************* BOTTOM OF DATA *******************************
```

Figure 3-4 SDSF Display of JES2 Checker Exception

3.1.4 Job Execution Control

The JES2 Job Execution Control (JEC) provides z/OS native support (through the standard JCL) for a job scheduling scheme within JES2 that can be used by any z/OS user. One goal is to preserve the relationships between the steps of a multistep job when it is broken down into a group of single (or few) step jobs.

Another goal is to combine a set of jobs into a network of jobs with related dependencies. The principal entity that controls job execution within JEC is a job group. A job group is a set of specifications between a JOBGROUP and ENDGROUP statement. The group defines the execution sequencing of a group of jobs and the jobs that are submitted after the job group specification.

The JES3 Dependent Job Control (DJC) includes a function that is similar to the JES2 JEC. DJC was originally provided as a JES3 function for installations that required a basic batch job networking capability and found that the use of conditional JCL (which uses COND codes) was cumbersome.

Over the years, most installations found that they required a more robust batch planning, control, and monitoring capability with less manual intervention. Now, the use of batch scheduling products, such as IBM Tivoli Workload Scheduler, is prevalent.

3.1.5 Deadline scheduling

With z/OS 2.3, JES2 provides functions to hold jobs until a specified time and to make jobs more likely to start running by a specified time. JES2 adds support for the SCHEDULE JCL statement with parameters HOLDUNTL and STARTBY. Use HOLDUNTL to hold the job until the specified time. Use STARTBY to move the job forward in the queue (increasing its priority, if needed) to make it more likely to start by the specified time.

The STARTBY function is also known as *deadline scheduling control*. The system cannot guarantee that a job finishes its execution by a certain time because too many variables are beyond the control of the system.

The system also cannot guarantee that a job begins its execution by a certain time. However, the system can now take measures so that the job has a fair chance to be the first in line to begin its execution by a specified time.

By using the STARTBY keyword on the SCHEDULE statement, users can specify an approximate time in the future when they want the job to start.

The system manages the priority of the job so that, by a target time, the job is near the top of the relevant job class or service class queue. In a sense, this function provides a time-controlled alternative to traditional priority aging.

Deadline scheduling is a function in JES3 that gives an user the ability to submit a job at a certain priority level at a certain time or day. It was also intended for jobs that must run at a designated time or period (for example, weekly).

These functions worked without a scheduling package or manual operator intervention. However, these functions can be better handled and controlled by a batch scheduling product, such as IBM Tivoli Workload Scheduler for z/OS, including its features for critical path processing and Event Triggered Tracking.

As processing capacity increased over the years, users came to expect that their jobs run when they are submitted; therefore, this function is not as critical as it used to be. If the job includes specific resource dependencies or must run at a certain time for charge back reasons, that process can also be controlled by using different job classes.

Note: The use of DEADLINE scheduling does not guarantee that the job executes at the exact time that you want. Some installations might find this function manually intensive to replace.

3.1.6 Priority aging

Jobs are selected to run based on job class and the available initiators in that class, and based on a priority in that particular job class queue. Priority aging is used to help jobs that were submitted on a system with an insufficient number of initiators.

Periodically, as defined by the relevant parameter, if the job is still on the job queue, the priority of the job is increased. This process potentially gives the job a better chance of being selected by an initiator and was intended to ensure that low-priority jobs did not languish in the job queue forever, while higher priority jobs were continually selected ahead of them.

Both JESes feature mechanisms to increase the priority of a job in the input queue based on how long the job is there. Consider the following points:

- In JES2, the function can be controlled by:
  - Specifying a priority on the /*PRIORITY JECL statement for JES2-managed initiators.
  - The use of the PRTYHIGH=, PRTYLOW=, and PRTYRATE= keywords on the JOBDEF initialization statement.
- In JES3, the function can be controlled by using SAGER/SAGEL and MAGER/MAGEL keywords on the SELECT INIT statement in the initialization deck

However, many installations use WLM-managed initiators. Workload management can dynamically manage the number of batch initiator address spaces in a JES2 or JES3 environment. You can selectively turn over control of the batch initiator management to WLM for one or more job classes. WLM starts new initiators as needed to meet the performance goals of this work.

By specifying MODE=WLM on the JES2 JOBCLASS statement or the JES3 GROUP statement, you indicate that the initiators for the job class are WLM-managed. In this environment, the JES priority of the job is irrelevant after it is selected for processing. Before it is selected for processing, the JES priority can be changed, which might change the designated service class for that job.

3.1.7 Support to eight-character job class name

JES2 supports up to eight-character job class names, such as JES3. The keyword CLASS= on JCL JOB card accepts the eight-character job class. With this function, JES2 and JES3 have similar capabilities that are related to job selection and the job class characteristics that can be assigned to a job. Also, to avoid many 2 - 8 character job class names being associated with a single initiator or a device, you can create job class groups to manage these associations.

3.1.8 Disk Reader Support

With z/OS 2.4, JES2 provides function similar to the JES3 disk reader DSP. It allows copying of a member from a predefined concatenation of PDSs, PDSEs, and z/OS UNIX directories to an internal reader, passing the records in the member to JES2 input processing.

This feature is implemented using three constructs that were introduced in z/OS 2.4:

- SUBMITLIB: Defines the libraries where the source members is stored. These libraries can be concatenated.
- SUBMITRDR statement: Defines the default for the input device. It is similar to the INTRDR statement.
- $SUBMIT command: Reads the members and passes them to the input processing.

3.1.9 JES2 Multi Job NJE Job Streams

Starting with z/OS 2.4, JES2 NJE was updated to support multiple jobs in a single job stream. Before z/OS 2.4, JES2 allowed only job per NJE job stream. If a second job was encountered, all jobs in the stream fail.

With this support, you can send a set of jobs to a node and have them processed by input processing in the correct order.

3.1.10 JES2 //*ROUTE XEQ Support

With z/OS 2.3, the support for many JES3 JECL statements was introduced. z/OS 2.4 adds the support for the JES3 //*ROUTE XEQ JECL. The syntax rules are the same as JES3, but the alternative form /*ROUTE XEQ is not supported because of conflicts with the JES2 /*ROUTE XEQ statement.

3.2 JES2-only functions

Several functions and features are unique to JES2 or behave differently in JES3. In this section, we describe some of these functions.

3.2.1 Job correlator

The job correlator (JOBCORR) is a 64-byte token that uniquely identifies a job to JES. The JOBCORR value is composed of a 32-byte system portion, which ensures a unique value, and a 32-byte user portion, which helps identify the job to the system. The UJOBCORR parameter of the JOB card specifies this 32-byte user portion of the job correlator. This job correlator provides the following features:

- Provides a larger name space for jobs (in addition to classical job name).
- Helps relating jobs to output and other records.
- Provides a simple way for applications to determine the Job ID of a job that was submitted.
- Is available with the z/OSMF REST API.

The UJOBCORR value can be overridden when the job is submitted by using the appropriate JES2 exits.

The job correlator is used to identify the job in multiple interfaces, including:

- JES operator commands
- ENF messaging
- Subsystem interfaces such as extended status and SAPI
- SMF records

In the following example, the user portion of the job correlator is set to JMAN_COMPILE:

```
//TEST JOB 333,STEVE,UJOBCORR=’JMAN_COMPILE’
```

Later, this value is combined with the system portion of the correlator to form a job correlator similar to the following example:

```
J0000025NODE1...C910E4EC.......:JMAN_COMPILE

|<-system portion----------------------->||<-user portion--------------->
```

In JES3 environments, this UJOBCORR parameter is accepted but ignored.

3.2.2 SPOOL migration

SPOOL migration allows an installation to quickly move data off a SPOOL volume in a period of minutes, instead of the days that a drain command takes. The processing can be done with active address spaces still accessing the volume.

The goal of the command is to get the source data set moved to a new volume or merged onto an existing SPOOL volume. The internal representation of the volume remains after it is merged onto an existing volume and persists until all jobs that were using the volume are purged. The volume continues to be displayed in **$D SPOOL** commands and in the volume list of a **$DJQ,SPOOL** command. The status of the “remnant” volume becomes MAPPED.

The two forms of SPOOL migration are MOVE and MERGE. In a move migration, the JES2 takes one INACTIVE SPOOL volume and moves it to a new volume that is not part of the SPOOL configuration. If three SPOOL volumes are available before a move, the JES2 continues with three SPOOL volumes after the move.

For a move, the source volume must be INACTIVE (HALTED). A merge migration takes the data on a SPOOL volume (in any state) and merges in into contiguous space on a target volume. If a merge starts with three volumes before the merge, it ends up with two volumes after the merge.

The third volume is displayed, but it is not being used (it is considered mapped). Merge is the least restrictive process. Any source volume can be merged to an appropriate target volume.

3.2.3 Encrypting and compressing spool data sets

Customer-sensitive data must be handled carefully to meet the demands of a secure world. IBM introduced in z/OS 2.3 DFSMS the capability to encrypt MVS data sets with no change to the applications. This support did not address spool data sets that can also contain sensitive data. In z/OS 2.4, JES2 can encrypt spool data sets by using z System hardware without application changes.

The data sets that are encrypted are also compressed, which provides storage efficiency at the same time. With this function, you can encrypt in stream and sysout data sets that are on spool.

This function also provides the capability to compress spool data sets without encrypting them.

3.2.4 JES2 Policies

Many installations must customize their JES2 system to achieve their requirements. These customizations involve coding JES2 exits that use the Assembler language. It also requires detailed knowledge of JES2 processing flow and control blocks.

In z/OS 2.4, JES2 introduces JES2 policies, which are a way to customize JES2 processing without the need to code exits. It is an alternative method to customize JES2 processing. Creating policies does not require programming skills. Customers formulate the required customizing in high-level terms that are based on the job requirements and attributes. JES2 policies do not replace JES2 exits that are still supported.

Multiple types of JES2 policies are available; however, this support was introduced in z/OS 2.4 and the initial support contemplates only some policy types. Many more types will be supported in the future as the function evolves.

3.2.5 Instruction Execution Protection

Instruction Execution Protection (IEP) is a feature of the IBM z14® hardware. It allows storage to be allocate storage in a non-executable state. This feature helps to protect systems from errors, such as stack overflows and malicious attacks.

JES2 uses this feature by changing the default behavior of the $GETMAIN macro. The default behavior in z/OS 2.3 was EXECUTABLE=YES. With z/OS 2.4, the default is EXECUTABLE=NO. This default can change the way some code that is written by users or vendors behaves; when code attempts to run inside storage allocated with this service, a program check occurs.

By using this service, JES2 assists in keeping systems more secure. If you upgrade to z/OS 2.4 and you are running on hardware previous to z14, JES2 continues to function the way it does with previous releases.

3.2.6 z/OSMF asynchronous job notifications

The z/OS jobs REST interface provides a set of REST services that allow a HTTP client application to perform operations with batch jobs and receive synchronous job notifications on a z/OS system. Through the z/OS jobs REST interface services, an application can perform the following tasks:

- Obtain the status of a job
- List the jobs for an owner, prefix, or job ID
- List the spool files for a job
- Retrieve the contents of a job spool file
- Submit a job to run on z/OS
- Cancel a job
- Change the job class
- Delete a job (cancel a job and purge its output)

The z/OS jobs REST interface services can be started by any HTTP client application that is running on the z/OS local system or a remote system (z/OS or non-z/OS). The z/OS jobs REST interface services are described in *IBM z/OS Management Facility Programming Guide*.

You can use the asynchronous job notifications function of z/OSMF to allow your programs to be notified when submitted jobs complete. With this function, the program that submits the job through the z/OS jobs REST interface services PUT method specifies a URL when submitting the job. When the job ends, z/OSMF returns an HTTP message to the URL location, indicating the job completion status. The data returned is in the form of a JSON document.

The key requirement is that you must create a subscription to the Common Information Model (CIM) jobs indication provider for your system. Also, if the job notifications require a secure network connection, you must enable an SSL connection between the client application and the server, including the sharing of digital certificates.

3.3 JES3-only functions

In this section, we describe JES3 functions and features that are unique to JES3 and are not available in JES2. Most of these functions are directly related to the way JES3 manages jobs.

3.3.1 Data Set Name disposition conflict resolution

JES3 resolves Data Set Name (DSN) conflicts before running a job. Consider the following sequence of conditions:

1. A job is submitted and requests access to a data set that is inconsistent with another job that uses that data set.
2. The newly submitted job is not selected for execution until the data set is freed by the running job.
3. Meanwhile, the job is placed in the JES3 allocation queue.

For example, if the new job requests exclusive access to a data set (DISP=OLD or DISP=MOD), and that data set is in use by another job, the new job does not start running.

Operators can display the JES3 queues by entering an **`*I S`** command. If jobs are in the allocation queue, you can determine why they are in the queue by entering a form of the **`*I S A J=nnnn`** command.

During job execution, a job might request allocation of a data set that is in use. In this case, the behavior of the JESs is the same and you receive messages similar to the messages that are shown in Example 3-1.

Example 3-1 Messages issued for data set enqueue conflict

```
IEF861I FOLLOWING RESERVED DATA SET NAMES UNAVAILABLE TO jobname
IEF863I DSN = data.set.name jobname RC = 04 RSN = 00000000 FROM SERVICE ENQ
IEF099I JOB jobname WAITING FOR DATA SETS
```

In JES2, the data set needs of a job are not considered when JES2 decides whether a job is selected for execution. As a result, conflicting data set enqueue and the “waiting for data set” message can occur more often.

However, you can use the JCL parameter DSENQSHR on JOB statement with the DSENQSHR subparameter of JOBCLASS definition. These parameters control how the system manages changes in data set disposition between job steps. In this way, you reduce control from exclusive to shared, which allows access by other jobs.

3.3.2 Spool partitioning

The JES3 spool can be divided into partitions that can be assigned to job classes, SYSOUT classes, or by user exit decision that is based on specific job characteristics. These partitions reserve spool space for important jobs processes.

To JES2, the spool is considered as a large repository space to hold the job data (input stream, SYSIN, and SYSOUT data set). This space can be used for any user who is authorized to process jobs. The spool partitioning on JES2 can be used as a way to assign specific spool volumes by creating a mask for users and the number of volumes on the FENCE parameter of the SPOOLDEF statement.

For this reason, to implement spool partitioning process on JES2, use the exits 11 and 12 to identify and control the spool volumes that a job can use. For more information, see Appendix E, “SPOOL partitioning exits sample code” on page 233.

Also, JES2 features a new enhancement for reserved spool space that is called *privileged space* that can be used for recovery purposes.

3.3.3 Job class group

A job class group is a named set of resource assignment rules to be applied to a group of job classes. System programmers define job class groups on JES3 initialization statements. They establish a link between a job class group and a job class by specifying a job class group name when they define the job classes. The job class group definitions in the initialization deck provide information about the resources that can be used by the set of jobs that is running.

The definition of job class group in JES2 is used to avoid the association of many 2 - 8 character job class names with a single initiator or a device. Then, you can create job class groups to manage these associations. In a manner similar to placing NJE nodes in SUBNETs, job classes can be defined to a job class group. Consider the following points:

- A job class can be in one job class group only, or in no job class group.
- A job class group is created when the first job class is added to the group.
- A job class group is deleted when the last job class is removed from the group.
- Deleting a job class also deletes the job class from its job class group.
- The maximum number of job class groups is 512 (in which case each group contains one job class).
- Job class group names must be unique, must range 2 - 8 alphanumeric characters, and must not match any existing job class name.

3.3.4 Printer naming conventions usage

JES3 does not restrict printer naming conventions. However, you must follow conventions in JES2 in assigning names to your printers. As a result, printer names in JES3 might be more meaningful to a human, but are not acceptable in JES2.

You might circumvent this issue by using JES2 destination IDs that match your old printer names. This issue probably has more effect on the operators because they must become familiar with the new printer names.

However, the output in JES3 is routed to writers that are then processed by printers. In JES2, output is routed to destinations and then printers select output that is associated with one or more destinations.

3.3.5 Main Device Scheduling

Main Device Scheduling (MDS) does not have much effect because all DASD volumes are always mounted and the tape drives are mostly virtual units. Nonetheless, JES3 provides the MDS feature to verify that all the resources (devices, volumes, and data sets) that are needed by a job are available before that job is executed. MDS can be disabled at a system level by using SETUP=NONE. It can still be overridden in jobs that specify //*MAIN SETUP= in their JCL.

Pre-execution setup

Pre-execution setup (JOB setup) is the basic feature of JES3 for pre-allocation of all devices, including DASD and tapes. JOB setup reserves all devices and mounts all volumes that are needed by a job before job execution.

Job setup can be requested on a job-by-job basis by specifying SETUP=JOB on the //*MAIN statement or on the JES3 initialization statement STANDARDS. SETUP=JOB is the default setting for the STANDARDS statement. Also, the resources are reserved from a JES3 setup perspective only. No ENQs or RESERVEs are issued.

For more information about MDS, see the chapter that is titled, “Main Device Scheduling”, in *ABCs of z/OS System Programming Volume 13*, SG24-7717.

Job setup also performs data set awareness. It prevents an initiator from being assigned if data set allocation conflicts exist. For more information, see 3.3.1, “Data Set Name disposition conflict resolution” on page 34.

The data set awareness feature is significant benefit in JES3, especially if you limit the initiators to a group. The feature stops a job from using an initiator and then waiting for data sets.

High water mark setup

The high water mark setup feature reduces the number of resources that are reserved for a job. The feature determines the maximum (or high water mark) number of devices that are used by any step in the job.

Consider the tape-drive requirements for the following example job that features three steps:

- Step One: Two tape drives
- Step Two: Three tape drives
- Step Three: One tape drive

This job reserves (or allocates) a total of three tape drives for the job because the maximum number of tape drives that is used by the job is three.

Without the high water mark setup feature, JES3 might attempt to allocate the total of six devices for the job. This allocation likely is not what was intended because JES3 views those devices as being unavailable to other jobs. This issue might by especially important for a long-running job in which only the last step requires many drives.

JES2 does not provide functions that are equivalent to MDS. You must take action before any migration to JES2 to eliminate use of JES3’s MDS features. It is likely that if you *do* use MDS, it is only used for tape.

If you use tape virtualization, it is reasonable to assume that you have more virtual tape drives than you ever use at one time. Therefore, disabling MDS probably has no visible effect on job throughput. Nevertheless, it is prudent to make this change before the migration. That way, if it does cause a problem, you can re-enable MDS while you investigate ways to address the problem.

3.3.6 JES3 device control and device fencing

The original default was for JES3 to control device allocation, including tape and DASD. Device fencing (also known as *device pooling*) was used to isolate or reserve a certain set of devices for a certain set of jobs or groups. For example, device fencing might ensure that a certain group of jobs uses DASD at a remote location only.

For DASD device allocation, it is now recommended to remove all devices from JES3 management by removing their definition from the JES3 Inish deck. This feature was most commonly used for tape drive allocations. However, with the combination of SMS-managed tape and tape virtualization, this issue is now less of a concern. Many customers no longer use JES3 to control their tape allocations.

You might be using JES3 MDS to control where a specific job is executed. For example, you might have a volume that contains a product that is licensed to only one system at a time. In this case, you can vary the volume online by using an operator command to only one system in the JESPlex and JES3 directs the job to that specific system. You can use Scheduling Environment to achieve the same effect in JES2.

Similar to member affinity, a job might be assigned a scheduling environment to ensure that it executes on specific members in the MAS. Use the SCHENV= keyword parameter on the JOB statement, or use the **`$T Job`** JES2 command.

Scheduling environments are installation-defined, 16-character names that might be available on any of the z/OS systems in the sysplex, or on none of the systems. Use workload management to define the scheduling environments and make them available or unavailable on each system based on the ON or OFF state of their resources.

Defining SCHENV for JES2

To get the JES3 job execution control based on specific resources in JES2, you can use the WLM Scheduling Environment function. With Scheduling Environment, you can control job execution based on resources and direct work to specific z/OS images where these resources are available.

Consider the following example:

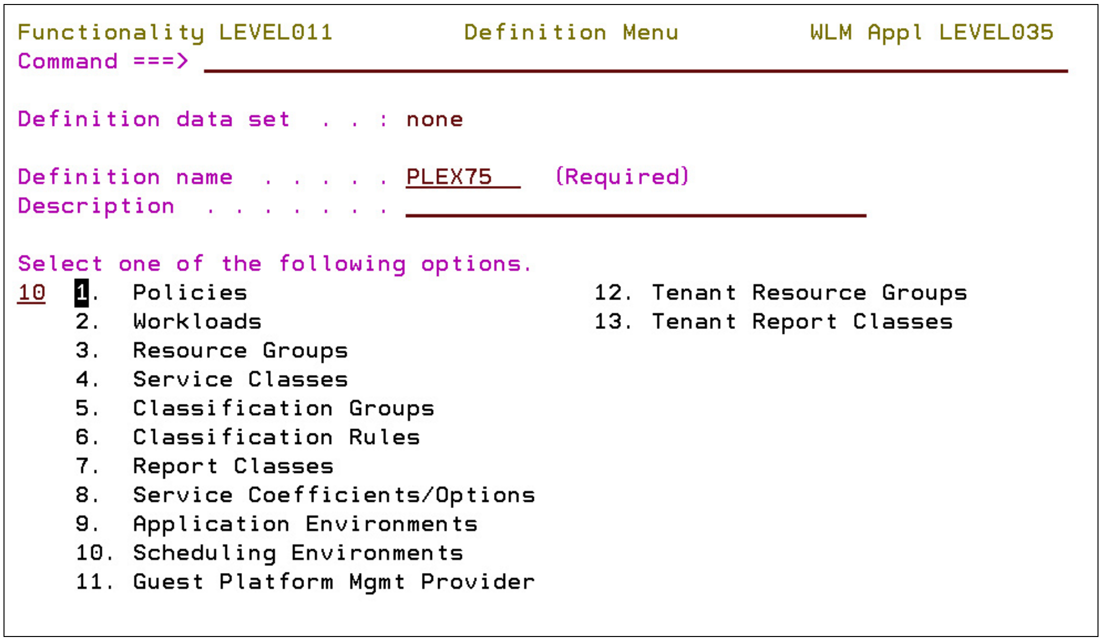
- You implemented the utilization of a DUMMY data set in JES3.
- You set DISP=OLD to serialize the execution of jobs (JES3 does not select a job if one of the resources that are required by the job is not available).

In this case, you can instead use Scheduling Environment in JES2 to accomplish the same objective.

The following example shows how to create, in JES2, the environment to perform the same serialization that JES3 does. We select JOBCLASS B to be the class that is used by the jobs to be serialized. We associate to this class the DD_DUMMY Scheduling Environment and restrict the number of WLM initiated initiators to one.

The following example uses only JES2 and WLM definitions. No special code or exits are required.

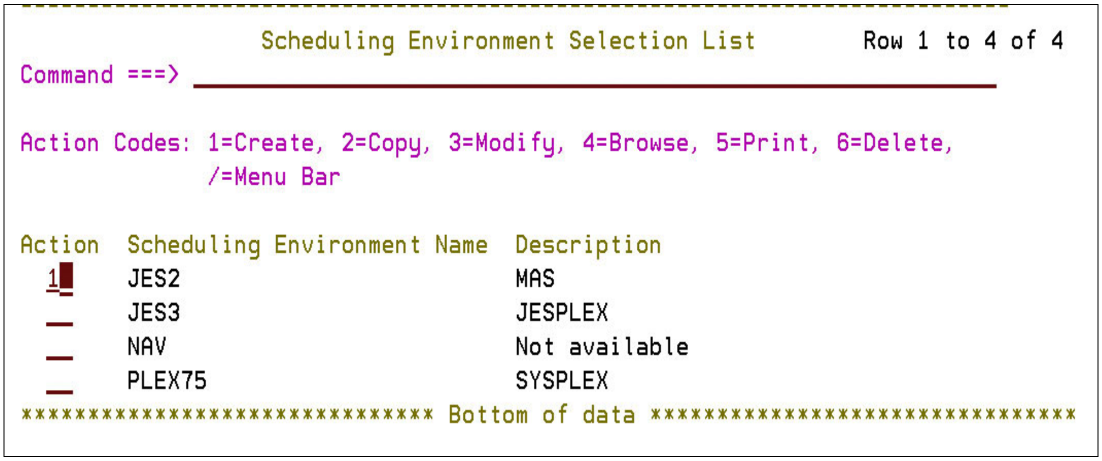
Figure 3-5 shows the WLM Definition Menu that is used to select the WLM definition that must be performed. In our case, we must define the Scheduling Environment. Select option 10 in the menu.



```
Functionality LEVEL011         Definition Menu         WLM Appl LEVEL035
Command ===> ____________________________________________________________

Definition data set  . . : none

Definition name  . . . . . PLEX75      (Required)
Description  . . . . . . . ________________________________

Select one of the following options.
10   1.  Policies                             12. Tenant Resource Groups
     2.  Workloads                            13. Tenant Report Classes
     3.  Resource Groups
     4.  Service Classes
     5.  Classification Groups
     6.  Classification Rules
     7.  Report Classes
     8.  Service Coefficients/Options
     9.  Application Environments
     10. Scheduling Environments
     11. Guest Platform Mgmt Provider
```

Figure 3-5 Accessing the WLM Definition Menu

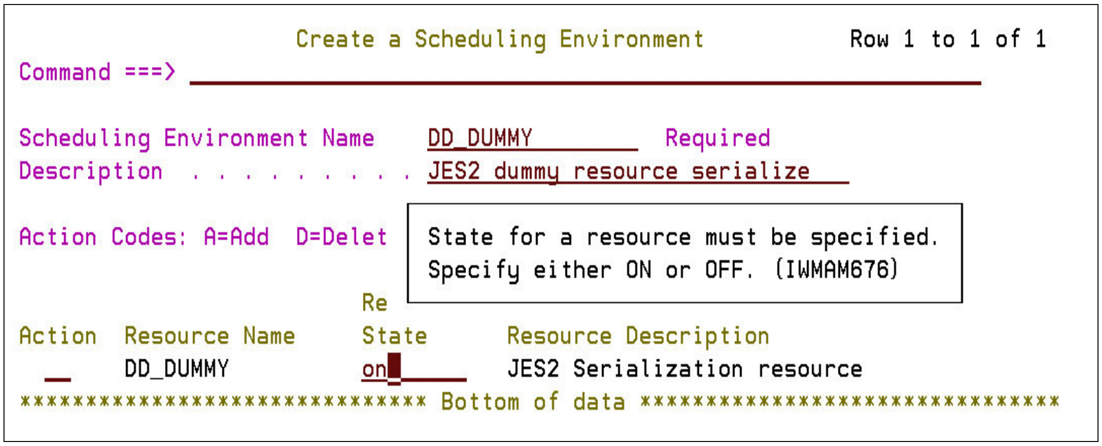
After you select the Scheduling Environments option, a new panel is displayed with all Scheduling Environments that are defined in the WLM policy. From this panel, you can select any element from the list and use **option 1=Create to create a new Scheduling Environment**, as shown in Figure 3-6.



```
                 Scheduling Environment Selection List       Row 1 to 4 of 4
Command ===> ____________________________________________________

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Scheduling Environment Name  Description
  1_    JES2                         MAS
  __    JES3                         JESPLEX
  __    NAV                          Not available
  __    PLEX75                       SYSPLEX
******************************* Bottom of data ********************************
```

Figure 3-6 WLM panel to create a Scheduling Environment

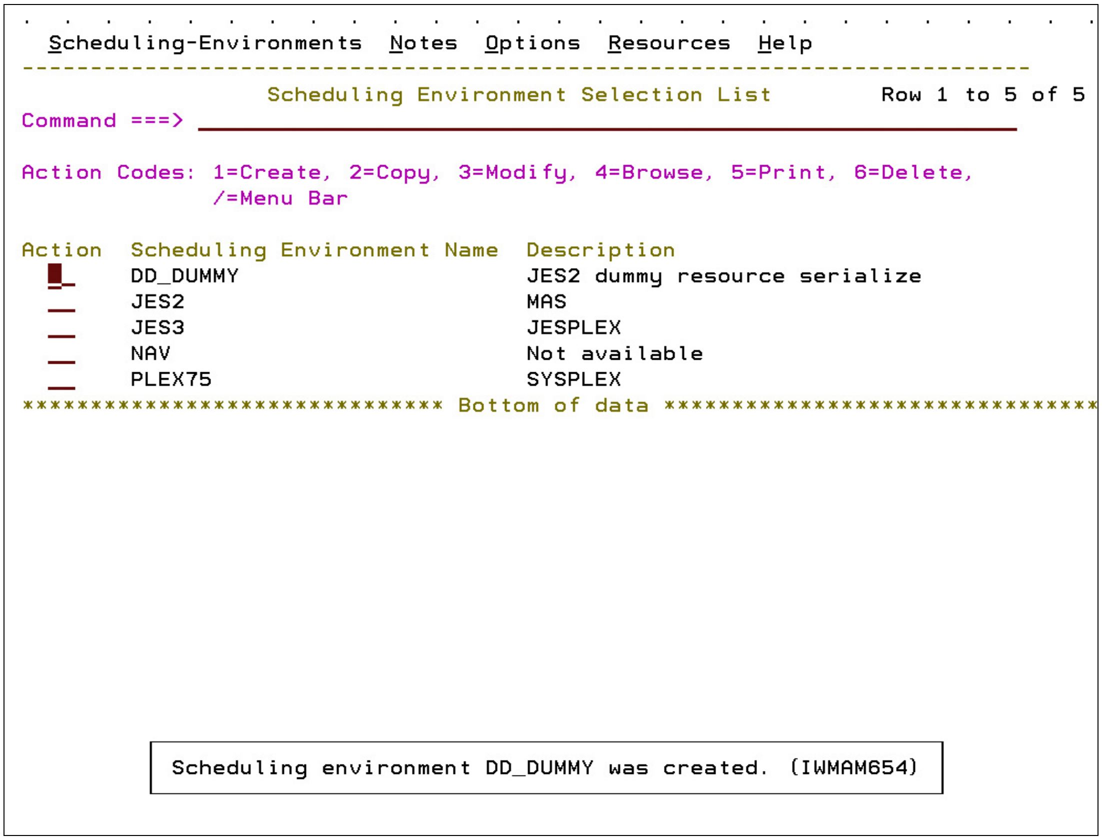
After you create your Scheduling Environment (as shown in Figure 3-7), you must create a resource that is associated with this Scheduling Environment. In this example, the Resource Name that is defined features the same name as the Scheduling Environment.



```
                    Create a Scheduling Environment        Row 1 to 1 of 1
Command ===> ____________________________________________________________

Scheduling Environment Name    DD_DUMMY________   Required
Description . . . . . . . . .  JES2 dummy resource serialize__

Action Codes: A=Add  D=Delet  State for a resource must be specified.
                              Specify either ON or OFF. (IWMAM676)
                         Re
Action  Resource Name    State      Resource Description
  __    DD_DUMMY         on______   JES2 Serialization resource
******************************* Bottom of data ********************************
```

Figure 3-7 Creating the DD_DUMMY Scheduling Environment with DD_DUMMY resource

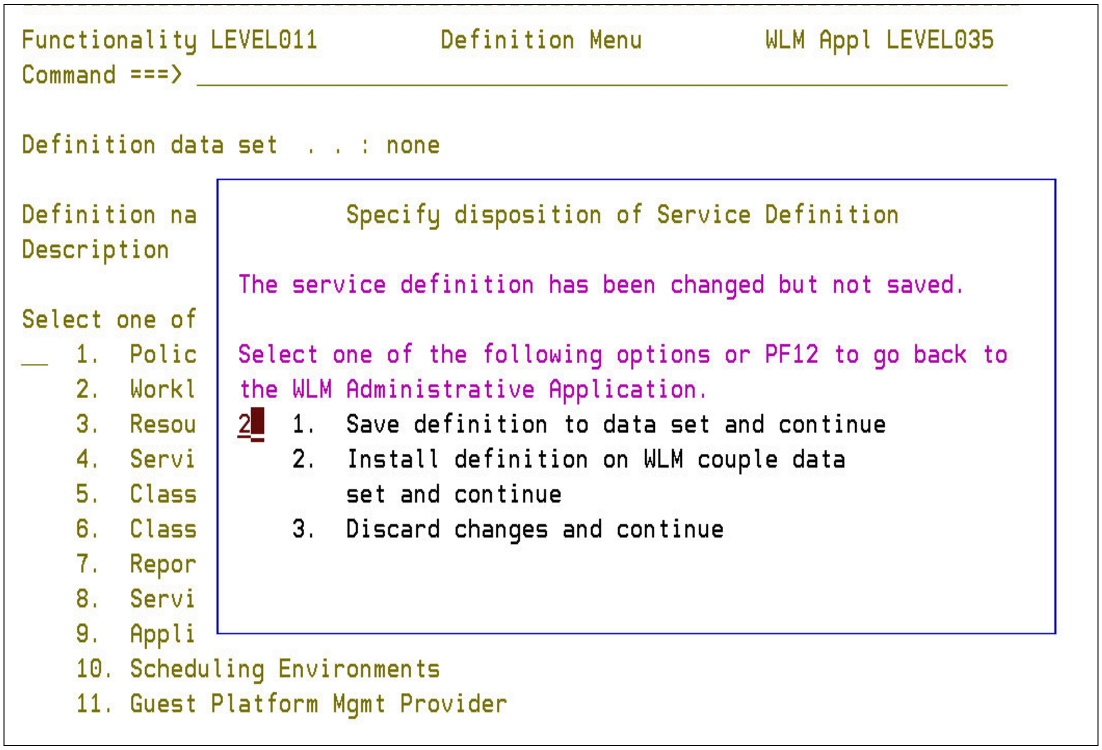
The Schedule Environment DD_DUMMY is created, as shown in Example 3-8.



```
 Scheduling-Environments  Notes  Options  Resources  Help
--------------------------------------------------------------------------
                  Scheduling Environment Selection List        Row 1 to 5 of 5
Command ===> ____________________________________________________________

Action Codes:  1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
               /=Menu Bar

Action  Scheduling Environment Name  Description
  __    DD_DUMMY                     JES2 dummy resource serialize
  __    JES2                         MAS
  __    JES3                         JESPLEX
  __    NAV                          Not available
  __    PLEX75                       SYSPLEX
******************************* Bottom of data ********************************

          Scheduling environment DD_DUMMY was created. (IWMAM654)
```

Figure 3-8 DD_DUMMY Scheduling Environment is created

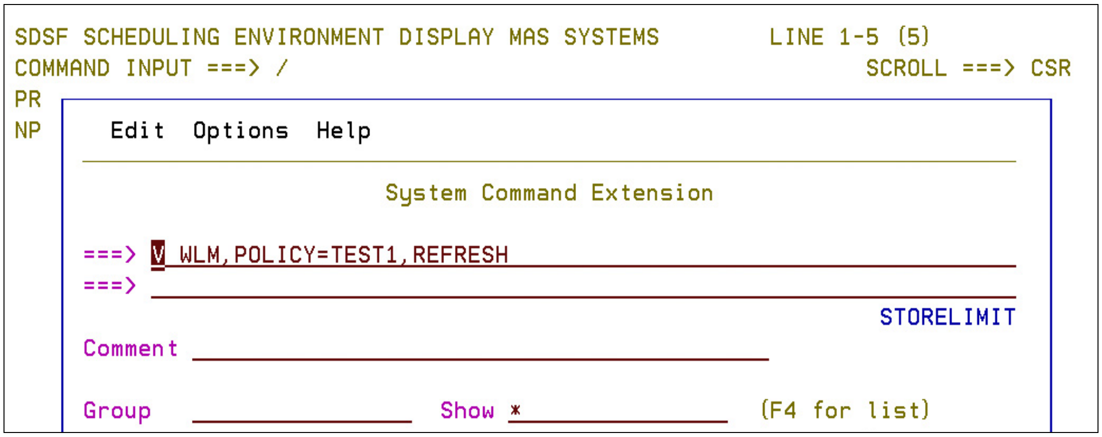
After the Scheduling Environment is successfully created, you must save the new policy into the WLM couple data set so that it is activated by the operator, as shown on Figure 3-9.



```
Functionality LEVEL011        Definition Menu        WLM Appl LEVEL035
Command ===> ____________________________________________________________

Definition data set  . . : none

Definition na  ┌──────────────────────────────────────────────────────────────┐
Description    │         Specify disposition of Service Definition            │
               │                                                              │
               │ The service definition has been changed but not saved.       │
Select one of  │                                                              │
__  1.  Polic  │ Select one of the following options or PF12 to go back to    │
    2.  Workl  │ the WLM Administrative Application.                          │
    3.  Resou  │ 2   1.  Save definition to data set and continue             │
    4.  Servi  │     2.  Install definition on WLM couple data                │
    5.  Class  │         set and continue                                     │
    6.  Class  │     3.  Discard changes and continue                         │
    7.  Repor  │                                                              │
    8.  Servi  │                                                              │
    9.  Appli  └──────────────────────────────────────────────────────────────┘
    10. Scheduling Environments
    11. Guest Platform Mgmt Provider
```

Figure 3-9   Installing the new definition on WLM couple data set

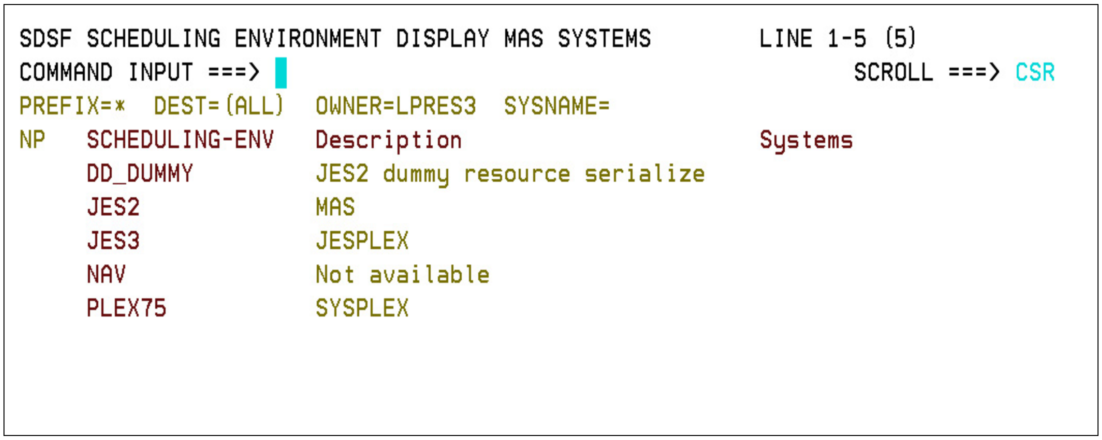
The activation of the WLM policy is done by issuing the **`Vary WLM,POLICY=`** operator's command from system console or SDSF panel, as shown in Figure 3-10.



```
SDSF SCHEDULING ENVIRONMENT DISPLAY MAS SYSTEMS        LINE 1-5 (5)
COMMAND INPUT ===> /                                       SCROLL ===> CSR
PR
NP     Edit  Options  Help
     ──────────────────────────────────────────────────────────────────
                          System Command Extension

     ===> V WLM,POLICY=TEST1,REFRESH
     ===>
                                                             STORELIMIT
     Comment

     Group                  Show *                 (F4 for list)
```

Figure 3-10   Activating the new definitions to WLM

By using the SDSF SE - Scheduling Environment panel (see Figure 3-11), you can display all Scheduling Environments that are defined for this Sysplex, and in which systems the Scheduling Environment is available.

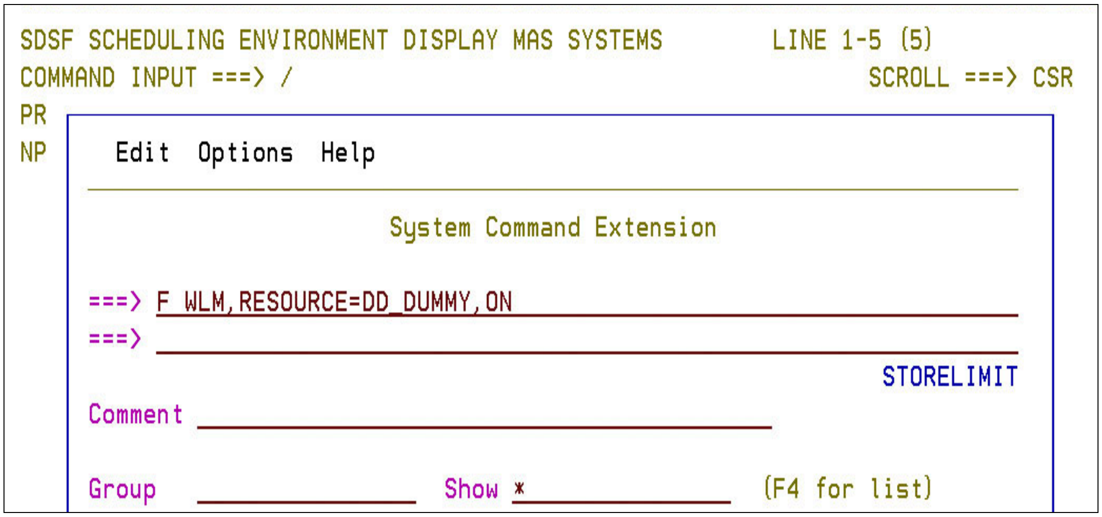


```
SDSF SCHEDULING ENVIRONMENT DISPLAY MAS SYSTEMS       LINE 1-5 (5)
COMMAND INPUT ===>                                           SCROLL ===> CSR
PREFIX=*  DEST=(ALL)  OWNER=LPRES3  SYSNAME=
NP   SCHEDULING-ENV   Description                      Systems
     DD_DUMMY         JES2 dummy resource serialize
     JES2             MAS
     JES3             JESPLEX
     NAV              Not available
     PLEX75           SYSPLEX
```

Figure 3-11   SDSF SE panel showing the defined Scheduling Environments

When the Scheduling Environment is defined or after an IPL, the resources are not available on any system in the sysplex and they must be activated. The initial status of the scheduling environments is shown in Figure 3-11.

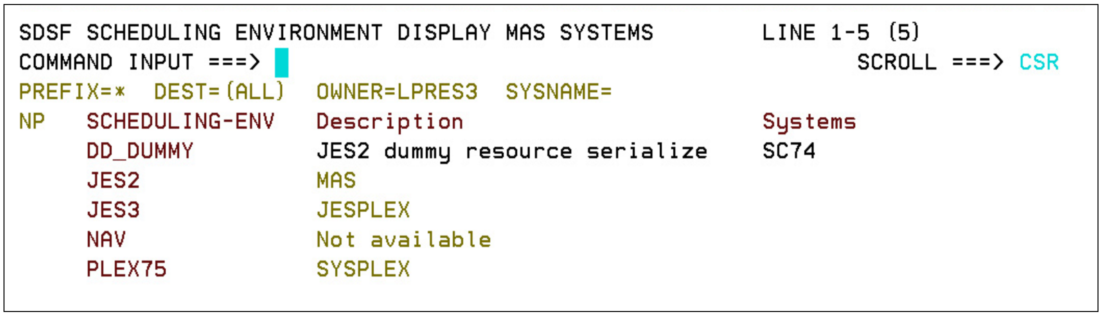
You activate a resource that is associated to a Scheduling Environment by issuing the `Modify WLM,RESOURCE=` operator’s command, as shown the Figure 3-12.



```
SDSF SCHEDULING ENVIRONMENT DISPLAY MAS SYSTEMS       LINE 1-5 (5)
COMMAND INPUT ===> /                                          SCROLL ===> CSR
PR
NP    Edit  Options  Help

                      System Command Extension

      ===> F WLM,RESOURCE=DD_DUMMY,ON
      ===>
                                                          STORELIMIT
      Comment

      Group                Show *                  (F4 for list)
```

Figure 3-12   Setting a resource that is associated to Scheduling Environment to ON

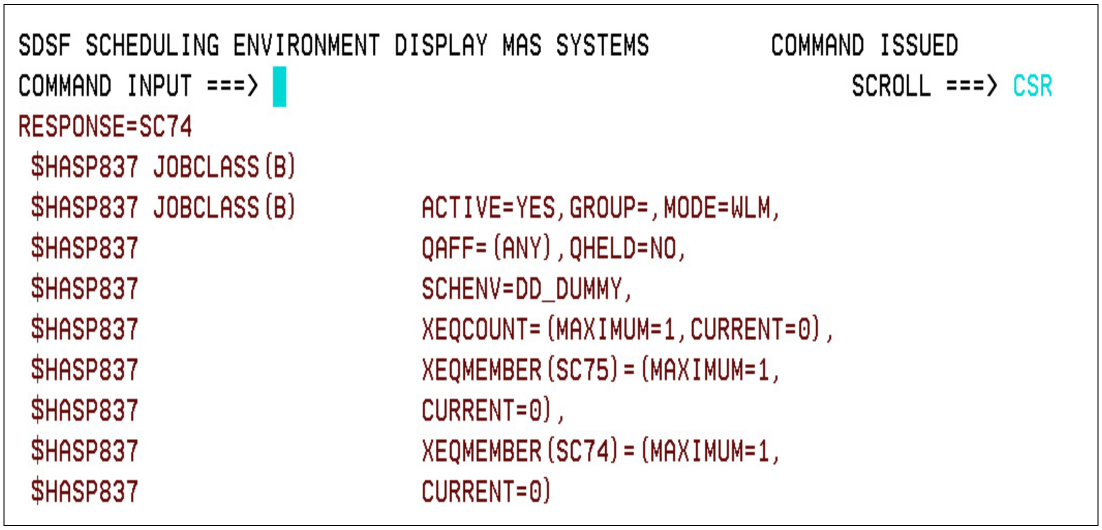
After the resource that is associated with the Scheduling Environment is activated on a system, it becomes available in that system. The jobs that are defined to that SCHENV can be selected by WLM to be executed in that z/OS image. The SDSF SE panel with the SCHENV=DD_DUMMY now available on system SC74 is shown in Figure 3-13.



```
SDSF SCHEDULING ENVIRONMENT DISPLAY MAS SYSTEMS       LINE 1-5 (5)
COMMAND INPUT ===>                                           SCROLL ===> CSR
PREFIX=*  DEST=(ALL)  OWNER=LPRES3  SYSNAME=
NP   SCHEDULING-ENV   Description                     Systems
     DD_DUMMY         JES2 dummy resource serialize   SC74
     JES2             MAS
     JES3             JESPLEX
     NAV              Not available
     PLEX75           SYSPLEX
```

Figure 3-13 SDSF SE panel displaying the Scheduling Environment available on system SC74

Consider a scenario in which you want to create an environment where only one job can be executed at a time. In addition to the Scheduling Environment creation, you must associate a specific job execution class to that Scheduling Environment (in our case, class B). You also must set the maximum number of jobs that can execute in the class to 1, as shown in Figure 3-14. The JOBCLASS(B) is set to MODE=WLM, SCHENV=DD_DUMMY and the XEQCOUNT=(MAXIMUM=1).



```
SDSF SCHEDULING ENVIRONMENT DISPLAY MAS SYSTEMS       COMMAND ISSUED
COMMAND INPUT ===>                                          SCROLL ===> CSR
RESPONSE=SC74
 $HASP837 JOBCLASS(B)
 $HASP837 JOBCLASS(B)        ACTIVE=YES,GROUP=,MODE=WLM,
 $HASP837                    QAFF=(ANY),QHELD=NO,
 $HASP837                    SCHENV=DD_DUMMY,
 $HASP837                    XEQCOUNT=(MAXIMUM=1,CURRENT=0),
 $HASP837                    XEQMEMBER(SC75)=(MAXIMUM=1,
 $HASP837                    CURRENT=0),
 $HASP837                    XEQMEMBER(SC74)=(MAXIMUM=1,
 $HASP837                    CURRENT=0)
```

Figure 3-14 Results of command $DJOBCLASS(B) showing the class that is associated to SCHENV

You can also use the same approach to direct JOBs to a specific z/OS image (for accounting purposes) and this JOBs uses a specific volume. In JES3, you can vary that volume online to that system only.

In JES2, you can associate those JOBs to a specific Schedule Environment and activate the resources of that environment in one z/OS image only (as we did for DD_DUMMY). In this situation, the initiator that handles the class can have XEQCOUNT=(MAXIMUM=) greater than 1.

4

JES2 functions to help migration

JES2 and JES3 evolved over time by introducing new functions that address the needs of their specific customer sets. As a result, specific statements in job entry control language (JECL) and job control language (JCL) are unique to JES2 or JES3 and created several differences between them.

This IBM Redbooks publication covers JES2 features that were included in JES2 that helps to decrease these differences. For more information about JECL and JCL the differences between the languages before z/OS V2.2, see *JES3 to JES2 Migration Considerations,* SG24-8083.

This chapter includes the following topics:

4.1 JES3 to JES2 migration options

We think that the best way to migrate from JES3 to JES2 is to convert your JES3 JCL and JECL to the JES2 equivalent instead of JES2 capability to interpret and convert JES3 JECL. This approach avoids confusion, eliminates the need to maintain skills in both types of JECL, and provides you with a clean base for moving forward. Nearly every JESS3-provided function can be re-created in a JES2 environment through a combination of standard JES2 functions, z/OS functions, and (if necessary) user exits.

4.2 Job Execution Control concept

The JES2 Job Execution Control (JEC) and Deadline Scheduling are features that were introduced in z/OS 2.2. We cover the following topics in this section:

- Purpose of the JEC
- New JOBGROUP and related JCL statements that comprise JEC
- Job group logging job, which is used to record state transitions within the group and facilitate job group management
- Simultaneous execution of a set of jobs that uses the CONCURRENT JCL statement
- Commands that are used to manage job groups
- Deadline Scheduling feature

JEC provides simple controls that can facilitate breaking down jobs into their constituent parts. That is, taking a multistep job and breaking it into multiple separate but related jobs. When these jobs are submitted, JES2 manages their execution in the correct order.

Also, by using JEC, you can define a set of two or more jobs for simultaneous execution. These jobs run in parallel on the same z/OS image. This function helps users that are running JES3 and JES2 by providing similar functions as the JES3 dependent job control (DJC) in the JES2 environment.

The principal entity that controls job execution within JEC is a job group. A job group is defined by a JOBGROUP JCL statement.

The following JCL statements provide JEC support:

- JOBGROUP: Creates a job group
- ENDGROUP: Denotes the end of the job group
- GJOB: Defines a job within a job group
- JOBSET: Provides convenient method to define and reference a set of jobs with identical dependencies
- SJOB: Defines a single job within the job set
- ENDSET: Denotes the end of the job set
- BEFORE: Defines jobs or job sets that the current job must run before
- AFTER: Defines jobs or job sets that the current job must run after
- CONCURRENT: Defines a set of jobs or job sets that must run at the same time (simultaneously) on the same z/OS image
- SCHEDULE: Associates a job with a job group

4.2.1 Job group concept

The job group is an entity that describes the relationships between multiple separate jobs. A job group is defined by the JOBGROUP JCL statement.

> **Note:** The job group definition defines the dependencies between the jobs only. The constituent jobs are defined separately by a traditional JCL statement.

The definition of a job group is static. Jobs cannot be added and dependencies cannot be changed after the job group is defined.

A job group includes a job group logging job that is associated with it. This job is a special type of job that acts as the front end for the job group. It serves the following purposes:

- The JESJCLIN data set of the logging job includes statements that are used to define the job group.
- The job log data set (JESMSGLG) contains messages about important events that are related to the jobs in the job group and to transitions in the job group state. For example, a message is logged when the job group completes, when each job starts, completes, and then is flushed.
- The logging job is used as a front end for the job group by the commands that act on the group (hold, cancel, and purge).
- The logging job is used as a front end for the job group by the extended status subsystem interface (SSI) and the job modify SSI. It is used for filtering, and so on.

After a job group is instantiated, jobs can then register to it through the SCHEDULE JCL statement. Any JES2 batch job can be registered to a job group. These concepts are shown in Example 4-1.

Example 4-1 JOBGROUP Example

```
//TESTE1 JOBGROUP
//*
//TEST1 GJOB
//*
//TEST2 GJOB
// AFTER NAME=(TEST1,TEST6)
//*
//TEST3 GJOB
// AFTER NAME=TEST1
//*
//TEST4 GJOB
// AFTER NAME=TEST2
//*
//TEST5 GJOB
// AFTER NAME=TEST2
//TEST6 GJOB
//TEST7 GJOB
// AFTER NAME=TEST6
//MYGROUP ENDGROUP
```

When this JCL is submitted, JES2 instantiates job group TESTE1 in the JES2 checkpoint. A logging job with the name TESTE1 is also created. Notice that no jobs are registered to the group at this point.

When jobs are registered, the following process occurs:

- TEST2 runs after TES1 and TEST6 complete.
- TEST3 runs after TEST1.
- TEST5 runs after TEST2 finishes.
- TEST7 runs after TEST6 completes.
- TEST1 and TEST6 have no dependencies (they run immediately).

The SCHEDULE JCL statement is used to register (associate) jobs with the job group (see Example 4-2).

Example 4-2 SCHEDULE JCL Statement

```
// SCHEDULE JOBGROUP=TESTE1
```

The SCHEDULE JCL must follow the JOB statement *before* the first EXEC statement. A JCL error is generated if it is misplaced.

After the JCL for a job with a SCHEDULE statement is successfully processed, the job is registered to the job group that is named on the JOBGROUP keyword. Jobs that are defined as part of a job group can be submitted in any order. However, you must submit them after the JCL of the job group is processed and the job group definition is committed to the JES2 checkpoint.

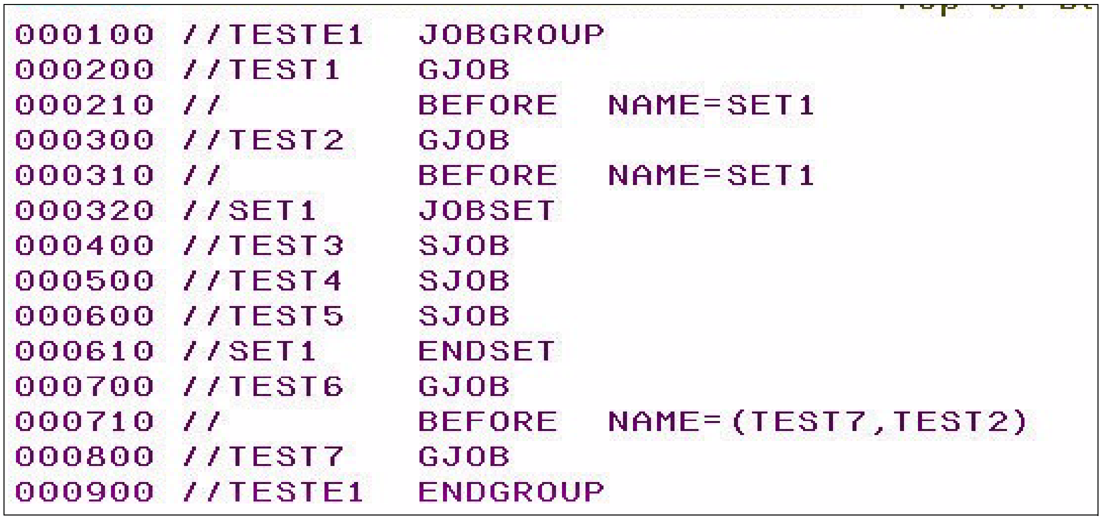
The job group owns certain resources and is authenticated by using the same process as normal batch jobs. This authentication includes checking profiles, such as the JESJOBS SUBMIT profile.

When a batch job is submitted that is registered to a job group, a check is made while the job is converting to validate the jobs access to the job group. If the user ID that owns the job group is the same as the user ID that owns the batch job, no other validation is performed (that is, no profiles are checked). If the user IDs are not the same, a check for authentication is made.

For more information about JOBGROUPs examples, see Appendix D, “DJC conversion and JEC examples” on page 225.

Use of JOBSET

JOBSET is a convenient method to define jobs with the same set of dependencies within a JOBGROUP. In the example that is shown in Figure 4-1 on page 47, TEST3, TEST4, and TEST5 share dependencies.



```
000100 //TESTE1   JOBGROUP
000200 //TEST1    GJOB
000210 //         BEFORE   NAME=SET1
000300 //TEST2    GJOB
000310 //         BEFORE   NAME=SET1
000320 //SET1     JOBSET
000400 //TEST3    SJOB
000500 //TEST4    SJOB
000600 //TEST5    SJOB
000610 //SET1     ENDSET
000700 //TEST6    GJOB
000710 //         BEFORE   NAME=(TEST7,TEST2)
000800 //TEST7    GJOB
000900 //TESTE1   ENDGROUP
```

Figure 4-1 Use of JOBSET

All references to the set are made by using the set name (SET1).

CONCURRENT statement

The CONCURRENT statement denotes that the following jobs must run simultaneously on the same z/OS image:

- The job that is specified by GJOB statement
- One or more jobs that are listed in the NAME parameter of the CONCURRENT statement

The jobs that are associated in this manner comprise what is called a *concurrent set*.

The syntax of the CONCURRENT JCL statement is shown in Example 4-4.

Example 4-3 CONCURRENT Statement

```
//TEST5 GJOB
// CONCURRENT NAME=name|(name,name,….)
```

It is important to understand the difference between the following aspects of job execution:

- Parallelism that is provided by the CONCURRENT statement
- Job-execution parallelism that is provided by the basic job group functionality

For example, consider jobs in a job group that do not have dependencies between them that are defined by BEFORE and AFTER JCL statements. Such jobs can run in any order on any z/OS image at the same time or at different times, depending on the operational state of z/OS images. In contrast to that example, jobs in a concurrent set must run at the same time on the same z/OS image.

JOBGROUP commands

Various operator commands can be used on the job group after the group is instantiated, including the following examples:

- $CG’MYGROUP’: Cancel a job group and all the jobs that are registered to it.
- $PG’MYGROUP’: Purge a job group (if completed) and all jobs that are registered to it.
- $HG’MYGROUP’: Hold a job group.
- $AG’MYGROUP’: Release a job group.
- $TG’MYGROUP’: Change attributes of a job group.

- $DG'MYGROUP',SUMMARY: Display a summary of a job group.
- $DG'MYGROUP',JOBS: Display only job information for the job group.

The example that is shown in Example 4-1 on page 45 is simple. The 10 new JCL statements provide the base that can be used to model complex job relationships.

The job log data set for the logging job shows step-by-step information about the execution flow of the jobs in the job group. Also, the JOBGROUP commands provide the control and monitoring functions that are needed to maintain a smoothly running job group.

4.2.2 Deadline scheduling

JES2 provides functions to hold jobs until a specified time and to make jobs more likely to start running by a specified time. Various keywords of a SCHEDULE statement provide the following convenient functions that add flexibility to the job management task:

- Use the **`HOLDUNTL;`** keyword to specify that the job must be in the held state until the time that is specified by the keyword. Then, the job is automatically released and can become eligible for execution.
- Use the **`STARTBY;`** keyword to specify the target execution deadline for the job.
- Use the **`WITH;`** keyword to indicate that the job must not run unless another (reference) job is active. When the job runs, the job must run on the same z/OS image as the reference job.

You can use these features on their own or together with the JEC to further enhance the job scheduling capabilities of native work management on z/OS.

HOLDUNTL

HOLDUNTL keyword on the SCHEDULE statement tells the system that a job must be placed in the held state at the submit time and be released at the specified time.

HOLDUNTL specifies the time in one of the following formats:

- Interval notation: Some number of hours and minutes from the time the job was submitted to the system. The syntax HOLDUNTL='+03:20 means that job must be released 3 hours and 20 minutes after the submission.
- Point In Time Notation: Direct specification of the time and optionally date when a job must be released. The syntax HOLDUNTL=('13:15','01/08/2018') means that job must be released at 1:15 PM on Aug. 5, 2018.

STARTBY

By using the STARTBY keyword on the SCHEDULE statement, a user can specify an approximate time in the future to start the job. The system manages the priority of the job so that the job is near the top of the relevant job class or service class queue by the target time. In a sense, this function provides a time-controlled alternative to traditional priority aging.

STARTBY syntax is identical to the syntax of the HOLDUNTL function. Target time can be specified in one of the same two formats: Interval or point-in-time notation.

WITH

Another job scheduling function is provided by the WITH keyword of the SCHEDULE statement. The WITH keyword indicates that a job must be selected for execution on the same system where another job (a reference job) is active. Until a reference job becomes active, the job that uses WITH function cannot be selected for execution.

The jobs can be submitted in any order. However, it is better to use the WITH keyword to submit and start the reference job before the jobs that point to it. A different order causes more system overhead.

A reference job does not have to be unique in the JESplex. If multiple jobs with the correct name are active on different z/OS images, the system chooses one of them. The choice of a z/OS image is unpredictable.

4.3 JES3 JECL processing support in JES2

JES2 can process various JES3 JECL statements if the corresponding options is activated. With this option, most of JES3 JECL statements can be processed by JES2 transparently. This option reduces the JECL conversion that is needed to run jobs that are originally coded for JES3 in a JES2 environment. As a result, you migrate JES3 to JES2 with minimal changes to your JES3 JECLs.

4.3.1 Activating JES3 JECL support

JES2 supports the processing of JES3 JECL statements in native support or translation into supported statements. The INPUTDEF JES3JECL and the JECLDEF JES3 initialization statements control how JES2 input processing handles various JES3 JECL statements. You can also use commands to perform the same control task.

Two levels of activation for this support are provided. In the first level, the following command enables the recognition of JES3 JECL syntax as JECL and not as a comment:

```
$T INPUTDEF,JES3JECL=PROCESS
```

This command tells JES2 that whenever a JES3 JECL statement is encountered, JES2 attempts to process it directly or by translating it into a JCL or a JES2 JECL statement.

However, if you issue the command **`$T INPUTDEF,JES3JECL=IGNORE`**, JES2 ignores all JES3 JECL statements. This option is the default.

The second level controls how each JECL statement is processed, as shown in the following example:

```
$T JECLDEF,JES3=(MAIN=PROCESS,DATASET=PROCESS,ROUTE=PROCESS,….)
```

This function supports the primary (//*) and alternative (/*) prefix for JES3 JECL. However, /* for NETACCT and ROUTE XEQ defaults to JES2 JECL.

INPUTDEF and JECLDEF feature single-member scope. Therefore, you can apply the same definition to all MAS members to keep consistent behavior among MAS members.

Also, in a hot start, INPUTDEF and JECLDEF in the initialization deck including defaults (IGNORE) are used unconditionally. This usage applies, even if you modified INPUTDEF or JECLDEF by **`$T`** commands before JES2 restarts. Therefore, you must define these statements explicitly in the initialization deck if you want to change the default value (IGNORE).

JES3 JECL toleration: JECLDEF for JES3

The following parameters are used for controlling JES3 JECL processing:

```
JECLDEF JES3=(
   MAIN          = PROCESS | IGNORE| WARN | FAIL
   FORMAT        = PROCESS | IGNORE| WARN | FAIL
   ROUTE         = PROCESS | IGNORE| WARN | FAIL
   OPERATOR      = PROCESS | IGNORE| WARN | FAIL
   DATASET       = PROCESS | IGNORE| WARN | FAIL
   ENDDATASET    = PROCESS | IGNORE| WARN | FAIL
   PROCESS       = PROCESS | IGNORE| WARN | FAIL
   ENDPROCESS    = PROCESS | IGNORE| WARN | FAIL
   NET           = PROCESS | IGNORE| WARN | FAIL
   NETACCT       = PROCESS | IGNORE| WARN | FAIL
   PAUSE         = PROCESS | IGNORE| WARN | FAIL)
```

Consider the following points:

- PROCESS means that the specific JES3 JECL statement is to be processed (translated or directly processed).
- IGNORE means that the specific JES3 JECL statement is not recognized and ignored, which is the default.
- WARN means that the specific JES3 JECL statement is to be processed (translated or directly processed), but a warning message is issued, as shown in the following example:

  ```
  HASP1130 JECL card xxxx encountered
  ```
- FAIL means that the specific JES3 JECL statement is not to be processed and the corresponding job is not run with a JCL ERROR, as shown in Example 4-4.

Example 4-4 FAIL indicating that specific JES3 JECL statement is not processed

```
IEFC452I jobname - JOB NOT RUN - JCL ERROR
$HASP106 JOB DELETED BY JES2 OR CANCELLED BY OPERATOR BEFORE EXECUTION
HASP1130 JECL card xxxx encountered
```

Level of support for JES3 JECL

Each statement has different support level, as shown in Example 4-5.

Example 4-5 Support levels for JES3 JECL

```
//*DATASET Tolerated, but not supported
//*ENDDATASET Required if //*DATASET used
//*FORMAT Partially supported (converted to OUTPUT JCL card)
//*MAIN Partially supported
//*NET Partially supported (converted to JES2 job group)
//*NETACCT Fully supported
//*OPERATOR Supported, but message text ends in 71, not 80
//**PAUSE Not supported, ignored if present
//*PROCESS Tolerated, but not supported
//*ENDPROCESS Tolerated, but not supported
//*ROUTE XEQ Fully supported
```

//*MAIN JECL keywords support

Each MAIN JECL keyword has different support level, as shown in Example 4-6.

*Example 4-6 //*MAIN keywords support level*

```
ACMAIN, IORATE, LREGION, MSS, RINGCHK, TRKGRPS, TYPE –Obsolete
BYTES, CARDS, CLASS, HOLD, JOURNAL, LINES, ORG –Supported
PAGES, PROC, SYSTEM –Supported
DEADLINE, EXPDTCHK, FAILURE, FETCH, SETUP, SPART –Not supported
THWSSEP, UPDATE, USER –Not supported
```

Obsolete means that a warning message is issued (as shown in the following example) and the keyword is ignored:

```
HASP1132 Obsolete keyword xxxx ignored
```

Not supported means that a warning message is issued (as shown in the following example) and the keyword is ignored:

```
HASP1133 Unsupported keyword xxxx used
```

//*FORMAT JECL keywords support

As with //*MAIN JECL keywords, each //*FORMAT JECL keyword has a different support level, as shown in Example 4-7.

*Example 4-7 //*FORMAT keywords support*

```
PR/PUpositional –ignored
DDNAME, CARRIAGE/FCB, CHARS, COMPACT, COPIES, DEST –supported
EXTWTR, FLASH, FORMS, MODIFY, PRTY, STACKER, TRAIN–supported
CHNSIZE, INT, OVFL, THRESHLD –not supported
```

Not supported means that an error message is issued (as shown in the following example) and the keyword is ignored:

```
HASP1133 Unsupported keyword xxxx used
```

Each //*FORMAT statement requires a //OUTPUT statement, which is placed just after the JOB statement and before the first EXEC statement and created automatically by JES2 as part of the //*FORMAT JECL processing. The name that is given the OUTPUT statements uses the format JES2*nnnn*, where *nnnn* begins at 0000, as shown in Example 4-8.

Example 4-8 OUTPUT statement format

```
HASP1312 JES20000 OUTPUT statement created for this //*FORMAT
//JES20000 OUTPUT DDNAME=SYSUT2,COPIES=2 <- created OUTPUT JCL by JES2
//*FORMAT PR,DDNAME=SYSUT2,COPIES=2 <- original JES3 //*FORMAT JECL
```

//*NETACCT keyword support

All keywords are supported in the same way that JES3 supports them.

//*NET JECL keywords support

Each //*NET JECL keyword has different support level, as shown in Example 4-9.

*Example 4-9 //*NET keywords support*

```
ID/NETID, ABCMP/AC,ABNORMAL,NORMAL,NETREL/NR –Supported
NHOLD/HC,NRCMP/PC,OPHOLD/OH,RELEASE/RL–Supported
DEVPOOL, DEVRELSE,RELSCHCT/RS -Obsolete
```

Obsolete means that a warning message is issued (as shown in the following example) and the keyword is ignored:

```
HASP1132 Obsolete keyword xxxx ignored
```

The following message is issued if keywords are supported:

```
HASP1309 //*NET card - Statement successfully processed
```

You see this message even when obsolete parameters are included with HASP1132.

For more information about DJC, see 6.6, “Transforming JES3 special functions” on page 143.

//*ROUTE XEQ keyword support

The keyword XEQ is supported in same way that JES3 supports it. JES2 does not support alternative JES3 /*ROUTE XEQ syntax because of a conflict with the JES2 /*ROUTE XEQ statement.

JES3 JECL support summary

JES3 JECL support is listed in Table , where it is assumed that INPUTDEF JES3JECL=PROCESS is enabled.

Table 4-1 JES3 JECL support

JES3 JECL	Support level	JECLDEF PROCESS	JECLDEF IGNORE	JECLDEF WARN	JECLDEF FAIL
DATASET	Tolerated, but not supported	No additional message	JCL error (IEFC019I[a])	HAS1130[b]	HASP1130[b] IEFC019I[a] $HASP106[c]
ENDDATASET	Tolerated, but not supported	No additional message	JCL error (IEFC019I[a])	HAS1130[b]	HASP1130[b] IEFC019I[a] $HASP106[c]
FORMAT	Partially supported (converted to JES2 job group)	HASP1312[d] (HASP1133[e])	N/A	HASP1130[b] HASP1312[d] (HASP1133[e])	HASP1130[b] $HASP106[c]
MAIN	Partially supported	No additional msg (HASP1132[f]) (HASP1133[e])	N/A	HASP1130[b] HASP1132[f]) (HASP1133[e])	HASP1130[b] $HASP106[c]
NET	Partially supported (converted to OUTPUT JCL)	HASP1309[g] HASP1300[h] HASP1301[i] HASP1304[j] (HASP1132[f])	N/A	HASP1130[b] HASP1309[g] HASP1300[h] HASP1301[i] HASP1304[j] (HASP1132[f])	HASP1130[b] $HASP106[c]

JES3 JECL	Support level	JECLDEF PROCESS	JECLDEF IGNORE	JECLDEF WARN	JECLDEF FAIL
NETACCT	Fully supported	No additional message	N/A	HASP1130[b]	HASP1130[b] $HASP106[c]
OPERATOR	Supported, but message text ends in 71, not 80	$HASP104[k]	N/A	HASP1130[b]	HASP1130[b] $HASP106[c]
PAUSE	Not supported, ignored if present	JCL error for //*PAUSE (IEFC607I)[l]	JCL error for //*PAUSE (IEFC607I[l])	JCL error for //*PAUSE (IEFC607I[l])	JCL error for //*PAUSE (IEFC607I[l])
PROCESS	Tolerated, but not supported	No additional message	N/A	HASP1130[b]	HASP1130[b] $HASP106
ENDPROCESS	Tolerated, but not supported	No additional message	N/A	HASP1130[b]	HASP1130[b] $HASP106[c]
ROUTE	Fully supported	No additional message	N/A	No additional msg	HASP6175[m] HASP6176[n] HASP6177[o]

a. IEFC019I MISPLACED statement STATEMENT
b. HASP1130 JECL card ccccccccccc encountered
c. $HASP106 jobname DELETED BY JES2 OR CANCELLED BY OPERATOR BEFORE EXECUTION
d. HAS1312 JES2nnnn OUTPUT statement created for this //*FORMAT
e. HASP1133 Unsupported keyword kkkkkkkk used
f. HASP1132 Obsolete keyword kkkkkkkk ignored
g. HASP1309 Job name //*NET statement successfully processed.
h. HASP1300 jobname registered to job group jobgroupname (SYSLOG only)
i. HASP1301 jobname in job group jobgroupname queued for execution (SYSLOG only)
j. HASP1304 job group jobgroupname is complete (SYSLOG only)
k. $HASP104 jobname text
l. IEFC607I JOB HAS NO STEPS
m. HASP6175 jobname Job to be transmitted has no records
n. HASP6176 jobname Expected JOB/NJB statement not found after //*ROUTE XEQ
o. HASP6177 jobname Encountered statement

JECL generic tracker

JES2 is instrumented to report the use of JES3 JECL in jobs that were submitted to the system and processed by JES2. Occurrences of JES3 JECL statements in a job stream are reported by using the Generic Tracker macro GTZTRACK. When GTZ tracking is enabled, JES2 records GTZ data that identifies the JES3 JECL statements that are found within a job stream. It applies to JES2 and JES3 JECL if the JECL type is active on INPUTDEF.

As an example of a job we created with JES3 JECL is shown in Example 4-10.

Example 4-10 JES3 JECL example

```
//TESTJOB JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//        REGION=0M
//*MAIN SYSTEM=SC75,TYPE=VS2,SETUP=HWS,LINES=(100,C)
//*FORMAT PR,DDNAME=SYSUT2,COPIES=1,THRESHLD=20000
//*NETACCT PNAME=FURUYA,BLDG=POK008
//STEP01  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
```

Tracking (that is, recording track events) is disabled by default. When tracking is disabled, invocations of GTZTRACK are allowed, but ignored by the system. Use the **SETGTZ TRACKING=ON** operator command to enable tracking, as shown in Example 4-11.

Example 4-11 Generic tracking

```
SETGTZ TRACKING=ON
GTZ1105I SETGTZ TRACKING PROCESSING IS COMPLETE
$HASP100 TESTJOB  ON INTRDR                            FROM TSU08997
LPRES1
IRR010I  USERID LPRES1   IS ASSIGNED TO THIS JOB.
ICH70001I LPRES1   LAST ACCESS AT 14:14:25 ON WEDNESDAY, JUNE 6, 2018
$HASP373 TESTJOB  STARTED - INIT 1    - CLASS A        - SYS SC75
IEF403I TESTJOB - STARTED - TIME=10.42.31
Jobname  Procstep Stepname  CPU Time      EXCPs     RC
TESTJOB  --None-- STEP01    00:00:00        313     00
IEF404I TESTJOB - ENDED - TIME=10.42.31
$HASP395 TESTJOB  ENDED - RC=0000
$HASP309 INIT 1    INACTIVE ******** C=ABCDE
D GTZ,TRACKDATA
GTZ1002I  10.42.59 GTZ TRACKDATA 392
FOUND 1 MATCHING TRACKED INSTANCE(S)
----------------------------------------------------------------------
 INSTANCE:      1                   COUNT:         1
 EVENTDESC:     '|01101000 0|00000000 0| INTRDR     TESTJOB  LPRES1 '+
                '  JES2       '
 OWNER:         IBMJES2             SOURCE:        HASCINJR
 EVENTDATA:     x0000000000000000   x0000000000000000
 PROGRAM:       *UNKNOWN            PROGRAMOFFSET: x0000000000000000
 HOMEJOB:       LPRES1              HOMEASID:      x0049
 EVENTJOB:      LPRES1              EVENTASID:     x0049
 AUTHORIZED:    YES                 FIRST TIME:    2018-06-07 10:42:30
```

The sample job specified three JES3 JECLs, as shown in Example 4-10 on page 53. In the same example, you can see the result of the **D GTZ,TRACKDATA** command, which includes the following displayed fields:

- OWNER: The string IBMJES2. It identifies the JES2 subsystem as the source of the GTZ record.
- SOURCE: Identifies the JES2 module that identified the occurrence of a JES3 control statement in the job stream (HASCINJR).
- EVENTDATA: Set to zeros.
- PROGRAM: Is *UNKNOWN.
- PROGRAMOFFSET: Is zeros because JES2 provides no program-specific information.
- EVENTDESC: A 46-character string in which JES2 provides information about the job stream and the JES3 control statement usage within the job stream. The contents of EVENTDESC by character position are listed in Table 4-2 on page 55.

Table 4-2 EVENTDESC field description

Position	Meaning and field description
1	A starting delimiter, which is the vertical bar character “l”, for the JES3 or JES2 JECL control statement usage indicators
2-18	Each character identifies whether a specific JES3 or JES2 JECL control statement is used in the job stream; Not used (0), Used (1)
2	//*DATASET statement
3	//*FORMAT statement
4	//*MAIN statement
5	//*NET statement
6	//*NETACCT statement
7	//*OPERATOR statement
8	//*PAUSE statement
9	//*PROCESS statement
10	Blank character
11	//*ROUTE statement
12	A delimiter that is the vertical bar character “l”
13	/*JOBPARM statement
14	/*MESSAGE statement
15	/*OUTPUT statement
16	/*ROUTE statement
17	/*SETUP statement
18	/*XEQ statement
19	/*NETACC statement
20	/*NOTIFY statement
21	Blank character
22	/*XMIT statement
23	A delimiter that is the vertical bar character “l”
24	Blank character
25-34	JES2 device name for point of entry for the job stream
35	Blank character
36-43	Job name.
44	Blank character
45-52	Submitting TSO user ID when SOURCE=HASCINJR
53	Blank character
54-57	JES2 subsystem name JES2 JECL

4.3.2 //*NET support detail

JES2 support for //*NET must be enabled by using the following commands:

```
$T INPUTDEF,JES3JECL=PROCESS
$T JECLDEF,JES3=(NET=PROCESS)
```

When enabled, JES2 migrates JES3 //*NET JECL to JES2 JOBGROUPs, called a //*NET JOBGROUP. Created JOBGROUPs are marked as including a //*NET statement origin that allows runtime processing to emulate the behavior of JES3 //*NET. However, runtime behavior differs from standard JOBGROUPs. JOBGROUP name is the NETID= as specified on the //*NET statement. Name space is shared with traditional JOBGROUPs.

//*NET JOBGROUPs are built dynamically as jobs with //*NET statements are processed during INPUT phase. For JEC JOBGROUPs, the entire network exists before any jobs are “registered” to it. The intent of this support is to emulate JES3 //*NET behavior as closely as possible.

HOLD counts are maintained by the //*NET JOBGROUP. Commands to modify HOLD counts are supported.

Provided commands are similar to the corresponding JES3 commands. Runtime behavior for //*NET JOBGROUPs is tailored to emulate JES3 //*NET behavior as much as possible. Substantial runtime differences exist with traditional JOBGROUP behavior. No requirement exists that dictates that a target RELEASE= job exist when a parent job runs. This condition is the same for NETREL= network or target job.

//*NET options and support available in JES2

The following JES2 //*NET options and support are available:

- NETID=name

  JES2 supports as the name of the JOBGROUP. A logging job is created.
- NHOLD=n

  JES2 supports this option.
- Release=(jobname, jobname, …)

  JES2 supports, treats similar to job group BEFORE processing.
- NETREL=(netid,jobname)

  JES2 supports this option.
- NORMAL=(D or F or R)

  JES2 supports this option.
- ABNORMAL=(D or F or R)

  JES2 supports this option.
- ABCMP=(KEEP or NOKP)

  JES2 supports this option.
- DEVPOOL=(ANY or NET or device-name,n)

  JES2 does not support (ignored).
- DEVRELSE=(YES or NO)

  JES2 does not support (ignored).

- NRCMP=(HOLD or NOHO or FLSH)

  JES2 supports this option.
- OPHOLD=(YES or NO)

  JES2 supports this option.
- RELSCHCT=n

  JES2 does not support (ignored).

//*NET and security

Because NETs use the job group infrastructure, some JEC processing applies. Logging job is created for NET job groups. The name of the logging job is the name in the NETID= keyword.

The owner of the logging job is the same as the job that triggered its creation. Jobs registering (connecting) to the job group must pass a security check. The profile for the job groups you want to protect must have the same user ID as the logging job or READ access to the JESJOBS entity by using the following format:

```
GROUPREG.nodename.groupname.userid
```

//*NET JOBGROUP peculiarities

//*NET JOBGROUPs are built dynamically as each job is processed. Dependencies are built from the RELEASE= statement of a “parent” job. However, the dependencies cannot be populated until the NORMAL= and ABNORMAL= definitions of a dependent job are processed. A window of time exists when a dependency is “undefined”.

//*NET dependencies are now initialized as undefined when they are created. Jobs in a //*NET JOBGROUP can run out of order. A job can run whenever the HOLD count reaches zero (by using a command or definition). That is, job execution is not fully controlled by dependencies. //*NET JOBGROUPs have no concept of a concurrent set of jobs.

//*NET and JOBGROUP commands

The following job group commands work against //*NET job groups:

- Display overview of a //*NET JOBGROUP:

  ```
  $DG*,JM=MYNET
  ```
- Display jobs in a //*NET JOBGROUP:

  ```
  $DG*,JM=MYNET,JOBS
  ```
- Display dependencies in a //*NET JOBGROUP:

  ```
  $DG*,JM=MYNET,DEP
  ```
- Cancel a //*NET JOBGROUP:

  ```
  $CG*,JM=MYNET
  ```
- Purge a //*NET JOBGROUP:

  ```
  $PG*,JM=MYNET
  ```

Note: MYNET is the NETID name (that is, - //*NET NETID=MYNET).Dependencies are created from the //*NET RELEASE=(jobname[,jobname]...) clause.

The following NHOLD operands are for jobs that are in //*NET JOBGROUPs:

- Display HOLD count value:

  `$DJ,JM=MYJOB,NHOLD`
- Decrement HOLD count value:

  `$TJ,JM=MYJOB,NHOLD=-`
- Increment HOLD count value:

  `$TJ,JM=MYJOB,NHOLD=+`
- Force HOLD count to zero:

  `$TJ,JM=MYJOB,NHOLD=0`

These commands are similar in function to what JES3 provides.

//*NET JOBGROUP peculiarities

Undefined dependencies result in pending or null data in displays. As shown in Example 4-12, successor (dependent) jobs (JOBB and JOBC) are not yet entered to JES2.

Example 4-12 JOBGROUP Display

```
$DG*,JM=NET1,JOBS,DEP
$HASP890 JOB(NET1) 651
$HASP890 JOB(NET1)      JOB GROUP JOB LIST
$HASP890                JOB NAME JOBID    JOB STAT COMP STAT HC
$HASP890                -------- -------- -------- --------- --
$HASP890                JOBZ     JOB09109 Q=HRDCPY COMPLETE   0
$HASP890                JOBC     NONE     NOT REG  PENDING    0
$HASP890                JOBB     NONE     NOT REG  PENDING    0
$HASP890                JOBA     JOB09107 Q=HRDCPY COMPLETE   0
$HASP890                JOB GROUP DEPENDENCY LIST
$HASP890                PARENT    DEP JOB   DEP STAT  COMP ACT
$HASP890                --------  --------  --------  ---------
$HASP890                JOBZ      JOBB      UNDEFINE  SATISFY
$HASP890                JOBA      JOBC      UNDEFINE  SATISFY
$HASP890                JOBA      JOBB      UNDEFINE  SATISFY
```

//*NET NETREL= support

//*NET JOBGROUPs supports NETREL=. This parameter reduces the NHOLD count for a job in another DJC network. If the target job group does not exist, a JOBGROUP object is created. A target job structure also is created (see Example 4-13).

Example 4-13 $DG example

```
$DG*,JM=NET2
$HASP890 JOB(NET2) 108
$HASP890 JOB(NET2)      JOB_GROUP_STATUS=PENDING,
$HASP890                ONERROR=SUSPEND,SYSAFF=(ANY),HOLD=(NO),
$HASP890                OWNER=LPRES1
$DG*,JM=NET2,JOBS
$HASP890 JOB(NET2) 110
$HASP890 JOB(NET2)      JOB GROUP JOB LIST
$HASP890                JOB NAME JOBID    JOB STAT COMP STAT HC
$HASP890                -------- -------- -------- --------- --
$HASP890                JOBD     NONE     NOT REG  PENDING   -1
```

Although another job released NET2 of JOBD, NET2 of JOBD is not yet entered to JES2.

//*NET JOBGROUPs and Extended Status SSI

JES2 JOBGROUPs are used to implement //*NET networks. Existing JOBGROUP Extended Status infrastructure is used without change. The RELEASE= job name list is returned as multiple dependency (STATDB) objects.

New job information (STATJQ) //*NET subsection is added (STATNETI), which includes the following //*NET statement keyword information:

- Original HOLD count value (STNEOHLD)
- NETREL= NETID name (STNENRID)
- NETREL= JOB name (STNENRJB)
- NORMAL= value (STNENORM)
- ABNORMAL= value (STNEABNR)
- ABCMP= value (STNEABCM)
- NRCMP= value (STNENRCM)
- OPHOLD= value (STNEPHLD)

The following job information (STATJQ) JOBGROUP in the information subsection (STATJZXC) was updated:

- Network Origin Indicator (STJZ1NOI):
  - OFF = Network is a static (JEC) JOBGROUP
  - ON = network is a //*NET JOBGROUP
- Current HOLD count value (STJZCHLD)
- NETREL= NETID name (STJZNRID)
- NETREL= JOB name (STJZNRJB)

Also, HOLD count filter is added (STATHCFV). This option allows filtering on current HOLD counts =, >, <, >=, <=, != to STATHCFV. See fields STATSHCE, STATSHCL, and STATSHCG for their dependencies on STATHCFV.

For more information, see Appendix F, “Alternative conversion programs” on page 263.

4.3.3 //*ROUTE XEQ support detail

JES2 implements the support to //*ROUTE XEQ JECL statement that reduces the work that is required to convert JECL statements in a JES3 to JES2 migration. With this support, all //*ROUTE XEQ JECL statements that are coded on JES3 JCL are recognized and processed by JES2.

The syntax rules used by JES2 are the same as JES3, as shown in the following example:

```
//*ROUTE XEQ dest[.vmguestid]
```

JES2 does not support the alternative /*ROUTE XEQ syntax that is used by JES3 because of conflicts with the JES2 /*ROUTE XEQ syntax.

The //*ROUTE XEQ statement is used to send the following input stream to a network node where the job is then run. JES2 stops transmitting the input stream records when it finds one of the following conditions:

- A second JOB statement after the //*ROUTE XEQ statement.
- The input stream runs out of records.

Statement after //*ROUTE XEQ

An error in the //*ROUTE XEQ statement can cause the JOB statement that is following the //*ROUTE XEQ to be processed at the submitting node. To prevent this issue, code NJB instead of JOB on the second JOB statement. JES2 changes the NJB to JOB before transmitting the job.

> **Note:** Consider the following points:
>
> - TSO/E users must code NJB instead of JOB on the second JOB statement.
> - If a JOB statement is not immediately following the //*ROUTE XEQ statement, the /*XMIT JCL statement must be used instead of //*ROUTE XEQ.

Example of //*ROUTE XEQ submitted by TSO/E user

In Example 4-14, the JOB JOBXEQ74 is sent to JES2 at system SC74 in node WTSCPLX7 by a SUBMIT command that is issued on a TSO/E edit session. The //*ROUTE XEQ statement tells JES2 to send the subsequent input stream (starting with the NJB statement JOBXEQ80) to the network node (WTSCPLX8) where the job is then run.

JES2 converts the NJB to JOB statement before transmitting the stream. Transmission of the input stream is stopped by the JOB statement JOBXEQ75. Job JOBXEQ80 is read and run by the system at node WTSCPLX8 (see Example 4-14). Job JOBXEQ75 is run at SC75 system of WTSCPLX7 node.

*Example 4-14 JOB using //*ROUTE XEQ JECL submitted from a TSO/E*

```
//JOBXEQ74 JOB (),'ITSO REDBOOKS',CLASS=B,MSGCLASS=X,
//         MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID
//*
//*ROUTE XEQ WTSCPLX8
//JOBXEQ80 NJB (),'ITSO REDBOOKS',CLASS=B,MSGCLASS=X,
//         MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID
//*
//OUTPUT   OUTPUT DEST=WTSCPLX7
//*
/*JOBPARM S=SC80
//*
//STEP01   EXEC PGM=IEFBR14
//*
//JOBXEQ75 JOB (),'ITSO REDBOOKS',CLASS=B,MSGCLASS=X,
//         MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID
//*
/*JOBPARM S=SC75
//*
//STEP01   EXEC PGM=IEFBR14
```

Example 4-15, Example 4-16 on page 61, and Example 4-17 on page 61 show the messages that are issued by the process of JOB by using the //*ROUTE XEQ.

Example 4-15 Messages from SC74 system processing the submitted job in Example 4-14

```
$HASP100 JOBXEQ74 ON INTRDR      ITSO REDBOOKS        FROM TSU02046
LPRES3
IRR010I  USERID LPRES3   IS ASSIGNED TO THIS JOB.
$HASP520 JOBXEQ80 ON L9.JT1
SE '10.11.58 JOB02052 $HASP526 JOBXEQ80 TRANSMITTED FOR EXECUTION AT
```

```
WTSCPLX8',LOGON,USER=(LPRES3)
$HASP100 JOBXEQ75 ON INTRDR      ITSO REDBOOKS          FROM TSU02046
LPRES3
IRR010I  USERID LPRES3   IS ASSIGNED TO THIS JOB.
$HASP524 L9.JT1     INACTIVE
$HASP250 JOBXEQ80 PURGED -- (JOB KEY WAS D654AE99)
$HASP540 JOBXEQ80 ON L9.SR1 FROM *UNKNOWN AT WTSCPLX8  65 RECORDS
SE '10.11.58 JOB02052 $HASP122 JOBXEQ80 (JOB02052 FROM WTSCPLX7)
RECEIVED AT WTSCPLX8',LOGON,USER=(LPRES3)
```

Example 4-16 Messages from SC75 processing the job JOBXEQ75

```
IEF196I IEFA111I INIT IS USING THE FOLLOWING JOB RELATED SETTINGS:
IEF196I          SWA=BELOW,TIOT SIZE=32K,DSENQSHR=DISALLOW,GDGBIAS=JOB
ICH70001I LPRES3   LAST ACCESS AT 09:07:53 ON WEDNESDAY, JUNE 26, 2019
$HASP373 JOBXEQ75 STARTED - WLM INIT  - SRVCLASS DFLT_MG  - SYS SC75
Jobname  Procstep Stepname  CPU Time       EXCPs     RC
JOBXEQ75 --None-- STEP01    00:00:00           8     00
$HASP395 JOBXEQ75 ENDED - RC=0000
SE '10.31.44 JOB02053 $HASP165 JOBXEQ75 ENDED AT WTSCPLX7 - JOBRC=0000
',LOGON,USER=(LPRES3)
```

Example 4-17 Messages from SC80 system processing the received job JOBXEQ80

```
$HASP373 JOBXEQ80 STARTED - INIT 1    - CLASS B         - SYS SC80
Jobname  Procstep Stepname  CPU Time       EXCPs     RC
JOBXEQ80 --None-- STEP01    00:00:00           8     00
$HASP395 JOBXEQ80 ENDED - RC=0000
$HASP309 INIT 1    INACTIVE ******** C=ABC
$HASP530 JOBXEQ80 ON L9.ST1  65 RECORDS
$HASP534 L9.ST1     INACTIVE
$HASP250 JOBXEQ80 PURGED -- (JOB KEY WAS D654AE99)
```

Example of //*ROUTE XEQ submitted by $SUBMIT command

In Example 4-18, the JOB statement JOBXEQ74 is sent to JES2 at system SC74 on node WTSCPLX7 by a $SUBMIT command that is issued from an operator's console. The //*ROUTE XEQ statement tells JES2 to send the subsequent input stream (starting with the JOB statement JOBXEQ80) to the network node (WTSCPLX8) where the job is then executed. Transmission of the input stream is stopped by the JOB statement JOBXEQ75. Job JOBXEQ75 is read and run by the system SC75 at node WTSCPLX7 (see Example 4-18).

*Example 4-18 JOB using //*ROUTE XEQ JECL submitted by a $SUBMIT Command*

```
//JOBXEQ74 JOB (),'ITSO REDBOOKS',CLASS=B,MSGCLASS=X,
//         MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID
//*
//*ROUTE XEQ WTSCPLX1
//JOBXEQ75 JOB (),'ITSO REDBOOKS',CLASS=B,MSGCLASS=X,
//         MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID
//*
//STEP01   EXEC PGM=IEFBR14
//*
//JOBXEQ76 JOB (),'ITSO REDBOOKS',CLASS=B,MSGCLASS=X,
//         MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID
//*
```

```
//STEP01   EXEC PGM=IEFBR14
//*
```

To submit the job that is shown in Example 4-18, the following command was issued at the operator's console:

```
$SUBMIT,MEMBER=JCLROUTE,DDNAME=SUBLIB00
```

This command calls the JES2 Disk Reader function to submit the job from a data set or an z/OS UNIX directory. The related messages from z/OS OPERLOG are shown in Example 4-19.

*Example 4-19 Sample of $SUBMIT command issued to start a JOB with //*ROUTE XEQ JECL*

```
$SUBMIT,MEMBER=JCLROUTE,DDNAME=SUBLIB00
$HASP000 OK
$HASP100 JOBXEQ74 ON INTRDR      ITSO REDBOOKS         FROM $SUBMIT
JCLROUTE
IRR010I  USERID LPRES3   IS ASSIGNED TO THIS JOB.
$HASP520 JOBXEQ80 ON L9.JT1
SE '10.49.56 JOB02054 $HASP526 JOBXEQ80 TRANSMITTED FOR EXECUTION AT
WTSCPLX8',LOGON,USER=(LPRES3)
$HASP100 JOBXEQ75 ON INTRDR      ITSO REDBOOKS         FROM $SUBMIT
JCLROUTE
IRR010I  USERID LPRES3   IS ASSIGNED TO THIS JOB.
$HASP524 L9.JT1     INACTIVE
$HASP250 JOBXEQ80 PURGED -- (JOB KEY WAS D654B715)
$HASP540 JOBXEQ80 ON L9.SR1 FROM *UNKNOWN AT WTSCPLX8  65 RECORDS
ICH70001I LPRES3   LAST ACCESS AT 10:31:44 ON WEDNESDAY, JUNE 26, 2019
$HASP373 JOBXEQ75 STARTED - WLM INIT  - SRVCLASS DFLT_MG  - SYS SC75
Jobname  Procstep Stepname  CPU Time      EXCPs     RC
JOBXEQ75 --None-- STEP01    00:00:00          8     00
$HASP395 JOBXEQ75 ENDED - RC=0000
SE '10.49.56 JOB02054 $HASP122 JOBXEQ80 (JOB02054 FROM WTSCPLX7)
RECEIVED AT WTSCPLX8',LOGON,USER=(LPRES3)
SE '10.49.56 JOB02055 $HASP165 JOBXEQ75 ENDED AT WTSCPLX7 - JOBRC=0000
',LOGON,USER=(LPRES3)
```

In Example 4-18 on page 61, the job JOBXEQ74 is used only to tell JES2 to transmit the subsequent job stream to WTSCPLX8 until the JOBXEQ75 JOB statement. The job JOBXEQ75 is processed locally.

The system SC80 messages when processing the job JOBXEQ80 that is transmitted by a //*ROUTE XEQ JECL statement on JOBXEQ74 job submitted by using the $SUBMIT command is shown in Example 4-20.

Example 4-20 Messages from SC80 system processing the transmitted job

```
$HASP100 JOBXEQ80 ON L9.JR1      ITSO REDBOOKS         FROM *UNKNOWN
AT WTSCPLX7
$HASP373 JOBXEQ80 STARTED - INIT 1    - CLASS B        - SYS SC80
Jobname  Procstep Stepname  CPU Time      EXCPs     RC
JOBXEQ80 --None-- STEP01    00:00:00          8     00
$HASP395 JOBXEQ80 ENDED - RC=0000
$HASP309 INIT 1    INACTIVE ******** C=ABC
$HASP530 JOBXEQ80 ON L9.ST1  65 RECORDS
```

```
$HASP534 L9.ST1     INACTIVE
$HASP250 JOBXEQ80 PURGED -- (JOB KEY WAS D654B715)
```

Error messages

The following error messages can be issued by JES2 when processing the //*ROUTE XEQ JECL statement:

- $HASP6175 JOBXEQ74: Job to be transmitted has no records
- $HASP6176 JOBXEQ74: Expected JOB/NJB statement not found after //*ROUTE XEQ
- $HASP6177 JOBXEQ74: Encountered //*

//*ROUTE XEQ password management option

To implement the //*ROUTE XEQ password management option, JES2 includes a NODE statement initialization parameter JES3_LOCAL_CHK that specifies whether (Yes) or not (No) batch jobs that are routed by the //*ROUTE XEQ JECL to this node should include the JOB card that is verified on the submitting node (Yes) or not (No).

This statement, combined with the PENCRYPT parameter, provides the same functions as the JES3 PWCNTL parameter on the NJERMT statement. The JES2 settings to match the JES3 PWCNTL specifications are listed in Table 4-3.

Table 4-3 JES2 definitions to match the JES3 PWCNTL specification

JES3 PWCNTL parameter	JES2 JES3_LOCAL_CHK parameter	PENCRYPT parameter
LOCALCHK	YES	N/A
SENDCLR	NO	NO
SENDENC	NO	YES

Setting of JES3_LOCAL_CHK parameter

The setting of JES3_LOCAL_CHK parameter can be done by defining the parameter on JES2 initialization data set or by issuing the **`$T NODE(xxxxxxxx)`** command.

The command that is issued to set the JES3_LOCAL_CHK parameter to node WTSCPLX8 to activate the local check of password on submitter job by using the //*ROUTE XEQ JECL statement is shown in Example 4-21.

Example 4-21 $T NODE command used to set the JES3_LOCAL_CHK parameter

```
$TNODE(WTSCPLX8),JES3_LOCAL_CHK=YES
$HASP826 NODE(9) 255
$HASP826 NODE(9)    NAME=WTSCPLX8,STATUS=(VIA/SC75),
$HASP826            AUTH=(DEVICE=YES,JOB=YES,NET=NO,SYSTEM=YES),
$HASP826            TRANSMIT=BOTH,RECEIVE=BOTH,HOLD=NONE,
$HASP826            PENCRYPT=NO,JES3_LOCAL_CHK=YES,
$HASP826            SIGNON=COMPAT,ADJACENT=NO,CONNECT=(NO),
$HASP826            DIRECT=NO,ENDNODE=NO,REST=0,SENTREST=ACCEPT,
$HASP826            COMPACT=0,LINE=0,LOGMODE=,LOGON=0,NETSRV=0,
$HASP826            OWNNODE=NO,PASSWORD=(VERIFY=(NOTSET),
$HASP826            SEND=(NOTSET)),PATHMGR=YES,PRIVATE=NO,
$HASP826            SUBNET=WTSCMXA,TRACE=NO
$HASP563 NODE(9)  NAME=WTSCPLX8  DEFINITION HAS CHANGED 335
         NODE(9)  JES3_LOCAL_CHK=NO CHANGED TO JES3_LOCAL_CHK=YES
```

With this option in effect, when submitting a job with a //*ROUTE XEQ JECL statement and USERID and PASSWORD set in JOB card, the authentication processing occurs on submitting node and, if validated, the job submitted by using //*ROUTE XEQ is considered a trusted job by that receiving node.

If the password is incorrect in the JOB card of the submitting job, the job fails. An error message is issued that states that the user ID or password cannot be verified. The target JOB is no transmitted. A verification failed occurrence on JOBXEQ74 that was used to transmit a job by using the //*ROUTE XEQ JECL card is shown in Example 4-22.

Example 4-22 Syslog messages to job JOBXEQ74 with invalid password

```
$HASP100 JOBXEQ74 ON INTRDR      ITSO REDBOOKS         FROM TSU02046
LPRES3
IRR010I  USERID LPRES3   IS ASSIGNED TO THIS JOB.
ICH408I USER(LPRES3  ) GROUP(SYS1    ) NAME(XXXXXXXXXXXXXXXXXXX ) 339
  LOGON/JOB INITIATION - INVALID PASSWORD
IRR013I  VERIFICATION FAILED. INVALID PASSWORD GIVEN.
```

However, when the JES3_LOCAL_CHK parameter is set to NO, as shown in Example 4-23, even if an incorrect password is coded in the JOB card, this password is not validated in the submitting system and the target job is transmitted.

Example 4-23 Setting of JES3_LOCAL_CHK and PENCRYPT parameters on JES2

```
$TNODE(WTSCPLX8),JES3_LOCAL_CHK=NO,PENCRYPT=YES
$HASP826 NODE(9) 342
$HASP826 NODE(9)    NAME=WTSCPLX8,STATUS=(VIA/SC75),
$HASP826            AUTH=(DEVICE=YES,JOB=YES,NET=NO,SYSTEM=YES),
$HASP826            TRANSMIT=BOTH,RECEIVE=BOTH,HOLD=NONE,
$HASP826            PENCRYPT=YES,JES3_LOCAL_CHK=NO,
$HASP826            SIGNON=COMPAT,ADJACENT=NO,CONNECT=(NO),
$HASP826            DIRECT=NO,ENDNODE=NO,REST=0,SENTREST=ACCEPT,
$HASP826            COMPACT=0,LINE=0,LOGMODE=,LOGON=0,NETSRV=0,
$HASP826            OWNNODE=NO,PASSWORD=(VERIFY=(NOTSET),
$HASP826            SEND=(NOTSET)),PATHMGR=YES,PRIVATE=NO,
$HASP826            SUBNET=WTSCMXA,TRACE=NO
$HASP563 NODE(9)  NAME=WTSCPLX8  DEFINITION HAS CHANGED 359
         NODE(9)  PENCRYPT=NO CHANGED TO PENCRYPT=YES
         NODE(9)  JES3_LOCAL_CHK=YES CHANGED TO JES3_LOCAL_CHK=NO
```

The messages that are issued for the job that is submitted from SC74 system on WTSCPLX7 with a user ID and an incorrect password that were coded in the JOB card is shown in Example 4-24. No password validation is done by the submitter node and the job is successfully transmitted to the destination node.

*Example 4-24 Job submitted using the //*ROUTE XEQ with JES3_LOCAL_CHK set to NO*

```
$HASP100 JOBXEQ74 ON INTRDR      ITSO REDBOOKS         FROM TSU02046
LPRES3
IRR010I  USERID LPRES3   IS ASSIGNED TO THIS JOB.
$HASP520 JOBXEQ80 ON L9.JT1
SE '11.51.50 JOB02061 $HASP526 JOBXEQ80 TRANSMITTED FOR EXECUTION AT
WTSCPLX8',LOGON,USER=(LPRES3)
$HASP524 L9.JT1     INACTIVE
$HASP250 JOBXEQ80 PURGED -- (JOB KEY WAS D654C4EC)
```

```
$HASP540 JOBXEQ80 ON L9.SR1 FROM *UNKNOWN AT WTSCPLX8  13 RECORDS
SE '11.51.50 JOB02061 $HASP122 JOBXEQ80 (JOB02061 FROM WTSCPLX7)
RECEIVED AT WTSCPLX8',LOGON,USER=(LPRES3)
```

However, the user ID that is used to submit the job is not propagated to the destination node. For this reason, the resulting job does not include a user ID that is associated to it and a security violation occurs because the user ID cannot be found in the destination node.

The Example 4-25 show the JOBLOG for the submitted job in the destination node.

Example 4-25 Job with Security Error on receiving node without user ID

```
J E S 2  J O B  L O G  --  S Y S T E M  S C 8 1  --  N O D E  W T S C P L X 8

11.51.50 JOB02061 ---- WEDNESDAY, 26 JUN 2019 ----
11.51.50 JOB02061  ICH408I USER(        ) GROUP(        ) NAME(???                )
                     LOGON/JOB INITIATION - USER AT TERMINAL          NOT RACF-DEFINED
11.51.50 JOB02061  IRR012I  VERIFICATION FAILED. USER PROFILE NOT FOUND.
$HASP106 JOB DELETED BY JES2 OR CANCELLED BY OPERATOR BEFORE EXECUTION
------ JES2 JOB STATISTICS ------
            9 CARDS READ
           13 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
            0 SYSOUT SPOOL KBYTES
         0.00 MINUTES EXECUTION TIME
```

4.4 Privileged support

In a JES3 environment, users can reserve spool space for emergency jobs by using spool partitioning. However, it is not so easy for JES2 to implement this function because you need exits 11 and 12, as described in Appendix E, “SPOOL partitioning exits sample code” on page 233.

However, JES2 provides the privileged support function that assists the system programmer in the resolution of critical JES2 resource shortage conditions. This function reserves a certain amount of critical resources for privileged job (STC, TSU, and JOB) use. This reserved resources can then be used by privileged jobs to diagnose and correct resources shortages. Privileged jobs enter the system by using an emergency subsystem.

A small percentage of SPOOL, jobs, output elements, and BERTs are set aside for privileged jobs. This approach assures that you have enough resources to log on, perform analysis, submit jobs, and resolve the root cause of resource exhaustion. Privileged resources can be used by privileged jobs, STCs, and TSO logons only.

Consider the following points:

- Its sole purpose is to provide the possibility to analyze and solve critical resource shortages.
- Its purpose is *not* to run high-priority workloads.

The following resources are guarded with this privileged support:

- BERTs
- JQEs
- JOEs
- SPOOL/Tracks

You must activate this function by using the following command:

```
$T LIMITS,PRIV=ON
```

When you successfully activate the function, the following message is displayed:

```
$HASP1401 Privilege Resource Support activated for -- <resource type>
```

You receive one message for each of the resources; however, if the activation for the specific resource fails, the following message is displayed:

```
$HASP1403 Privilege Resource Support could not be activated for <resource type>
```

This failure occurs when the free elements of the resource are smaller than the minimum number required (see Table 4-4). The required free element numbers come from the default setting or from the small environment.

The required free elements for each resource in default environment are listed in Table 4-4.

Table 4-4 Privileged resources in default environment

Resource	Free required to activate	Number reserved for privilege	Maximum
BERTs	20,000	1% of free	756
JOEs	20,000	1% of free	600
JQEs	10,000	1% of free	300
SPOOL(TGs)	20,000 TGs	400 TGs	400

However, these requirements might be too large for small environments. In this case, you can run the following command to activate a “small environment,” which has smaller requirements:

```
$T LIMITS,PRIV=ON,SMALLENV=ON
```

The required free elements for each resource in a small environment are listed in Table 4-5.

Table 4-5 Privileged resources in small environment

Resource	Free required to activate	Number reserved for privilege	Maximum
BERTs	1,200	150	150
JOEs	600	60	60
JQEs	100	10	10
SPOOL(TGs)	380 TGs[a]	45 TGs per MAS member	32*45

[a] Also required for activating SPOOL small environment, including Track Groups (TGs): If the product of (45 TGs) X (number of MAS members) exceeds 12.5% of total free TGs, then activation cannot occur and the $HASP1403 message is issued.

You can show the current LIMITS status in the default environment, as shown in Example 4-26.

Example 4-26 Sample display of privileged resources in a default environment

```
$DLIMITS
$HASP1490 LIMITS(1) 489
LIMITS(1)
PRIVILEGE SUPPORT IS ON
SPOOL PRIVILEGE SUPPORT IS ON
SPOOL UTILIZATION ON 11 JUN 2018 AT 16:35:25
----------  NON-PRIVILEGED  ---------|---  PRIVILEGED  --
    MAXIMUM   WARN%       IN-USE    %|  MAX  AVAILABLE
     39,617      80       8,889   22|  400        400
SPOOL EXHAUST: 23 JUL 2018 AT 09:02
*********************************************************
$HASP1490 LIMITS(2) 490
LIMITS(2)
PRIVILEGE SUPPORT IS ON
JQE PRIVILEGE SUPPORT IS OFF
JQE UTILIZATION ON 11 JUN 2018 AT 16:35:25
----------  NON-PRIVILEGED  ---------|---  PRIVILEGED  --
    MAXIMUM   WARN%       IN-USE    %|  MAX  AVAILABLE
      3,000      80         726   24|    0          0
JQE EXHAUST: 15 JUN 2018 AT 06:42
*********************************************************
$HASP1490 LIMITS(3) 491
LIMITS(3)
PRIVILEGE SUPPORT IS ON
JOE PRIVILEGE SUPPORT IS OFF
JOE UTILIZATION ON 11 JUN 2018 AT 16:35:25
----------  NON-PRIVILEGED  ---------|---  PRIVILEGED  --
    MAXIMUM   WARN%       IN-USE    %|  MAX  AVAILABLE
     10,000      80       1,528   15|    0          0
JOE EXHAUST: 9 JUL 2018 AT 04:37
*********************************************************
$HASP1490 LIMITS(4) 492
LIMITS(4)
PRIVILEGE SUPPORT IS ON
BERT PRIVILEGE SUPPORT IS OFF
BERT UTILIZATION ON 11 JUN 2018 AT 16:35:25
----------  NON-PRIVILEGED  ---------|---  PRIVILEGED  --
    MAXIMUM   WARN%       IN-USE    %|  MAX  AVAILABLE
      2,100      80         387   18|    0          0
BERT EXHAUST: 12 JUN 2018 AT 18:44
*********************************************************
```

You also can display the current LIMITS status in a small environment, as shown in Example 4-27.

Example 4-27 Sample display of privileged resources in a small environment

```
$DLIMITS
$HASP1490 LIMITS(1) 965
LIMITS(1)
PRIVILEGE SUPPORT IS ON,SMALL ENVIRONMENT IS ON
SPOOL PRIVILEGE SUPPORT IS ON
SPOOL UTILIZATION ON 13 JUN 2019 AT 11:20:56
----------  NON-PRIVILEGED  ---------|---  PRIVILEGED  --
    MAXIMUM   WARN%       IN-USE    %|  MAX  AVAILABLE
     39,927      80      13,428   34|   90         90
SPOOL EXHAUST: 19 AUG 2019 AT 07:51
*********************************************************
$HASP1490 LIMITS(2) 966
LIMITS(2)
PRIVILEGE SUPPORT IS ON,SMALL ENVIRONMENT IS ON
```

```
JQE PRIVILEGE SUPPORT IS ON
JQE UTILIZATION ON 13 JUN 2019 AT 11:20:56
----------  NON-PRIVILEGED  ---------|---  PRIVILEGED  --
    MAXIMUM   WARN%       IN-USE    %|  MAX  AVAILABLE
      2,990     80       1,380   46|   10        10
*********************************************************
$HASP1490 LIMITS(3) 967
LIMITS(3)
PRIVILEGE SUPPORT IS ON,SMALL ENVIRONMENT IS ON
JOE PRIVILEGE SUPPORT IS ON
JOE UTILIZATION ON 13 JUN 2019 AT 11:20:56
----------  NON-PRIVILEGED  ---------|---  PRIVILEGED  --
    MAXIMUM   WARN%       IN-USE    %|  MAX  AVAILABLE
       9,940     80        3,762   38|   60        60
 *********************************************************
 $HASP1490 LIMITS(4) 968
 LIMITS(4)
 PRIVILEGE SUPPORT IS ON,SMALL ENVIRONMENT IS ON
 BERT PRIVILEGE SUPPORT IS ON
 BERT UTILIZATION ON 13 JUN 2019 AT 11:20:56
 ----------  NON-PRIVILEGED  ---------|---  PRIVILEGED  --
     MAXIMUM   WARN%       IN-USE    %|  MAX  AVAILABLE
       1,950     80          444   23|  150       150
 *********************************************************
```

4.5 JES2 initialization data set checker

The JES2 Initialization Data Set Checker, similar to JES3 Initialization Stream Checker, allows installations to verify their initialization data sets without starting a JES2 subsystem. The process can detect syntax errors in initialization statements and problems with settings that might prevent JES2 from starting. The checker can verify that the statements are valid for a cold or warm start.

If you are verifying parameters on a warm start, you must run the checker within a SYSPLEX with an active member of the MAS. The checker uses XCF messaging to extract information from the active MAS member to achieve the following goals:

- Verify that the parameters are valid on a warm start
- Perform more analysis of resource usage

For more information about the JES2 initialization data set checker, see Chapter 5.2.1, “Verifying the JES initialization deck” on page 95.

4.6 SMF 84 record support

JES2 provides a function to track JES2 resource usage that is similar to JES3 JMF (JES3 Measurement Facility).

With this function, JES2 automatically writes SMF 84 records for resource monitoring, if you do not disable the recording for SMF 84 in IFASMFxx.

Record type 84 contains information that is collected by JES2 or JES3 monitors. This record is intended to provide insights into what the subsystems are doing during the interval that the record represents. Each record type 84 contains a common section (with header, product, and general information portions) and a subtype section unique for each record. JES2 creates subtype 21. Subtype 21 contains resource limit and usage information.

JES2 SMF 84 records include the following sections:

- Header: No changes from JES3 header
- Product section: Same mapping that JES3 uses
- General section: Section present, but nothing is set in this section
- Data section: Subtype 21 –JES2 resource usage:
  - Memory usage subsection (24-, 31-, and 64-bit areas) mapped by R84MEMJ2:
    - <16M USER
    - <16M SYSTEM
    - >16M USER
    - >16M SYSTEM
    - >2G PRIVATE
  - Resource usage subsection (limit, low, high, average, count over warn, and so on) reported by resource name mapped by R84RSUJ2:
    - BERT
    - BSCB
    - BUFX
    - CKVR
    - CMBS
    - CMDS
    - ICES
    - JNUM
    - JOES
    - JQES
    - LBUF
    - NHBS
    - SMFB
    - TBUF
    - TGS
    - TTAB
    - VTMB
    - ZJC

A 1,344-byte record is produced in each SMF interval.

Because no official formatting program exists for SMF 84 subtype 21, you must develop your own formatting program, depending on your requirements. For more information about a sample program to format SMF 84 subtype 21, see Appendix C, “Sample SMF84 Report program” on page 203. Although you can use this code as a starting point, you must thoroughly test the final code that you deploy.

4.7 Eight-character JOB CLASS and JOB CLASS GROUP support

In this section, we describe eight-character JOB CLASS and JOB CLASS GROUP support.

Eight-character JOB CLASS support

JES2 supports up to eight-character job class names, which is similar to the JES3 support. The JCL JOB card CLASS= parameter is expanded to support up to eight characters. Classes can be managed by using the **$ADD/$DEL JOBCLASS** command. Other commands are also updated to support eight-character job classes.

An eight-character JOB CLASS can be and assigned to an initiator, as shown in Example 4-28.

Example 4-28 Sample syslog to add and display eight characters job class

```
$ADD JOBCLASS(NEWADD),ACTIVE=YES
$HASP837 JOBCLASS(NEWADD) 200
$HASP837 JOBCLASS(NEWADD)    ACTIVE=YES,GROUP=,MODE=JES,
$HASP837                     QAFF=(ANY),QHELD=NO,SCHENV=,
$HASP837                     XEQCOUNT=(MAXIMUM=*,CURRENT=0),
$HASP837                     XEQMEMBER(SC75)=(MAXIMUM=*,
$HASP837                     CURRENT=0),
$HASP837                     XEQMEMBER(SC74)=(MAXIMUM=*,
$HASP837                     CURRENT=0)
-------------------------------------------------------------------------------
$TI(49),CLASS=(NEWADD)
$HASP892 INIT(49) 202
$HASP892 INIT(49)  STATUS=INACTIVE,CLASS=(NEWADD),NAME=49,
$HASP892           ASID=0066
```

You can specify the eight-character JOB CLASS, as shown in Example 4-29.

Example 4-29 Sample JCL to specify 8-character job class

```
//JOBA  JOB MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//        REGION=0M,CLASS=NEWADD
//STEP01  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=SYS1.PARMLIB(JES3IN00),DISP=SHR
//SYSUT2   DD DUMMY
//SYSIN    DD DUMMY
/*
```

4.7.1 JOB CLASS GROUP support

JES2 also supports job class groups, which is similar to the JES3 support. Each job class can be in one job class group. Job class group names and job classes must be unique and you cannot have a job class group with the same name as a job class.

As shown in Example 4-30, two job classes are defined, each belonging to one job class group.

Example 4-30 Sample SYSLOG to add and display job class group

```
$ADD JOBCLASS(TEST1),ACTIVE=YES,GROUP=GRP1
$HASP837 JOBCLASS(TEST1) 853
$HASP837 JOBCLASS(TEST1)     ACTIVE=YES,GROUP=GRP1,MODE=JES,
$HASP837                     QAFF=(ANY),QHELD=NO,SCHENV=,
$HASP837                     XEQCOUNT=(MAXIMUM=*,CURRENT=0),
$HASP837                     XEQMEMBER(SC75)=(MAXIMUM=*,
$HASP837                     CURRENT=0),
$HASP837                     XEQMEMBER(SC74)=(MAXIMUM=*,
$HASP837                     CURRENT=0)
-------------------------------------------------------------------------------
$ADD JOBCLASS(TEST2),ACTIVE=YES,GROUP=GRP1
$HASP837 JOBCLASS(TEST2) 918
$HASP837 JOBCLASS(TEST2)     ACTIVE=YES,GROUP=GRP1,MODE=JES,
$HASP837                     QAFF=(ANY),QHELD=NO,SCHENV=,
$HASP837                     XEQCOUNT=(MAXIMUM=*,CURRENT=0),
$HASP837                     XEQMEMBER(SC75)=(MAXIMUM=*,
$HASP837                     CURRENT=0),
$HASP837                     XEQMEMBER(SC74)=(MAXIMUM=*,
$HASP837                     CURRENT=0)
-------------------------------------------------------------------------------
$DCLASSGRP(GRP1)
$HASP816 CLASSGRP(GRP1)      TEST2,TEST1
-------------------------------------------------------------------------------
$TI(50),CLASS=(GRP1)
$HASP892 INIT(50) 254
$HASP892 INIT(50)  STATUS=INACTIVE,CLASS=(GRP1),NAME=50,
$HASP892           ASID=0065
```

A job class group facilitates selecting on job classes. Initiators and Offload Job Transmitters can specify 1 - 36 single character job classes or 1 - 8 multi- (or single-) character job classes or job class groups.

You can specify a job class name that is included in a job class group in the CLASS parameter of a JOB statement, as shown in Example 4-31. You cannot specify a job class group name directly in the CLASS parameter of a JOB statement.

Example 4-31 Sample JCLs to specify job class included job class group

```
//JOBA  JOB MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//        REGION=0M,CLASS=TEST1
//STEP01  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=SYS1.PARMLIB(JES3IN00),DISP=SHR
//SYSUT2   DD DUMMY
//SYSIN    DD DUMMY
```

```
/*
-------------------------------------------------------------------------------
//JOBB  JOB MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//        REGION=0M,CLASS=TEST2
//STEP01  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=SYS1.PARMLIB(JES3IN00),DISP=SHR
//SYSUT2   DD DUMMY
//SYSIN    DD DUMMY
/*
```

When an initiator that is assigned to a job class group selects a job for execution, it does so in a round-robin fashion. When a job is selected, the classes are rotated so the next selection starts with the next job class in the group.

For example, assume that you have ten jobs that specify CLASS=TEST1, and other ten jobs that specify CLASS=TEST2 in the job queue. In this example, TEST1 and TEST2 are included in a job class group GRP1. As a result, the initiator that GRP1 is assigned to select TEST1 jobs and TEST2 jobs in round-robin fashion.

The following updates were made to the CLASS= parameter in the command and initialization statement:

- CLASS=ABCD: Implies single-character job classes A, B, C, and D.
- CLASS=(ABCD): Implies four-character job class or job class group ABCD.

4.8 Interpreter after converter support

In a JES3 environment, the z/OS interpreter is called before a job is transferred to the initiator. However, in a JES2 environment, z/OS interpreter is normally called when the job is transferred to the initiator. If you want the same behavior in JES2 as you have in JES3, you can specify the following parameters:

- JOBDEF INTERPRET=JES (a parameter in the initialization statement)
- $T JOBDEF, INTERPRET=JES command

INTERPRET specifies when JES2 calls the z/OS interpreter to process a job.

With this option, the z/OS interpreter is called at the end of conversion processing. The use of this option includes the following benefits:

- Earlier detection of JCL errors that are detected by the MVS interpreter. This function allows errors to be detected, even if the job never runs f(TYPRUN=HOLD).
- Processing of JESDS OUTPUT statements to control data set attributes, even if the job never runs.

INTERPRET=JES also means that the converter and interpreter run in a JES2CI address space that is separated from a JES2 address space. Therefore, SYSZTIOT ENQ contention can be avoided between a spool offloaded allocation delay in JES2 address space and conversion JCL allocation in JES2CI address space. It is a best practice to use INTERPRET=JES.

INIT specifies to call the interpreter when the job is selected for execution by an initiator. Starting the interpreter in the initiator is the traditional JES2 processing method. This behavior is the default.

For example, the JCL that is shown in Example 4-32 immediately fails with JCL error in JES3 environment (even though TYPRUN=HOLD is requested) because the interpreter can find the error before the initiator.

Example 4-32 Sample JCL with TYPRUN=HOLD

```
//JOBA  JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//        REGION=0M,TYPRUN=HOLD
//STEP01  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=SYS1.PARMLIB(JES3IN00),DISP=XXX
//SYSUT2   DD DUMMY
//SYSIN    DD DUMMY
/*
```

However, in the JES2 default environment where JEBDEF INTERPRET=INIT, the job is held until released, as shown in Example 4-33. Therefore, timing for JCL error detection is much earlier in JES3.

Example 4-33 Sample job login INTERPRET=JES

```
14.32.52 JOB09571 ---- WEDNESDAY, 13 JUN 2018 ----
14.32.52 JOB09571  IRR010I  USERID LPRES1   IS ASSIGNED TO THIS JOB.
14.43.05 JOB09571  IEF452I JOBA     - JOB NOT RUN - JCL ERROR <- Held more than 10 minutes until released
14.43.05 JOB09571  $HASP396 JOBA     TERMINATED
------ JES2 JOB STATISTICS ------
  13 JUN 2018 JOB EXECUTION DATE
            8 CARDS READ
           25 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
            1 SYSOUT SPOOL KBYTES
         0.00 MINUTES EXECUTION TIME
        1 //JOBA  JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
          //        REGION=0M,TYPRUN=HOLD
          IEFC653I SUBSTITUTION JCL - CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=PRES1,REGION=0M,TYPRUN=HOLD
        2 //STEP01  EXEC PGM=IEBGENER
        3 //SYSPRINT DD SYSOUT=*
        4 //SYSUT1   DD DSN=SYS1.PARMLIB(JES3IN00),DISP=XXX
        5 //SYSUT2   DD DUMMY
        6 //SYSIN    DD DUMMY
          /*
 STMT NO. MESSAGE
-
        4 IEF643I UNIDENTIFIED POSITIONAL PARAMETER IN THE DISP FIELD
```

However, with JOBDEF INTERPRET=JES enabled, you can detect JCL errors in the early interpreter phase, as does JES3.

Enabling JOBDEF INTERPRET=JES requires the following conditions:

- z11 mode
- All members must be z/OS V2R1 or later

Also, specifying INTERPRET=JES causes EXIT 60 to be driven instead of EXIT 6. If you use EXIT 6, you might need to also provide similar function in EXIT 60 before setting INTERPRET=JES.

4.9 Functions similar to deadline scheduling

JES2 features two functions that are similar to JES3 deadline scheduling. The first function is a way to hold a job until a specific time. The other function is a way to move up a job's position in the execution queue over time so that it runs before a certain time.

4.9.1 HOLDUNTL on SCHEDULE JCL statement

HOLDUNTIL= indicates a date and time (or an amount of time that the job is held) when the job can be released from hold status.

You specify a date and time when a job can be released, as shown in Example 4-34. The job is released on 16:30 on Jun. 12 2019 in this example.

Example 4-34 Sample JCL to specify HOLDUNTIL for specific date and time

```
//JOBZ  JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//        REGION=0M
//SCHED SCHEDULE HOLDUNTL=('16:30','2019/163')
//STEP01  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=SYS1.PARMLIB(JES3IN00),DISP=SHR
//SYSUT2   DD DUMMY
//SYSIN    DD DUMMY
/*
```

Another option is to specify how long the job must be held, as shown in Example 4-35. The job is released after 30 minutes from the job submission in this example.

Example 4-35 Sample JCL to specify HOLDUNTIL for specific period

```
//JOBZ  JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//        REGION=0M
//SCHED SCHEDULE HOLDUNTL='+00:30'
//STEP01  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=SYS1.PARMLIB(JES3IN00),DISP=SHR
//SYSUT2   DD DUMMY
//SYSIN    DD DUMMY
/*
```

Until a job is released, $DJ shows the status as shown in Example 4-36.

Example 4-36 Sample display for the job before release

```
$DJ9498
$HASP890 JOB(JOBZ) 551
$HASP890 JOB(JOBZ)      STATUS=(AWAITING EXECUTION),CLASS=A,
$HASP890                PRIORITY=8,SYSAFF=(ANY),HOLD=(JOB,
$HASP890                HOLDUNTL)
```

4.9.2 STARTBY on SCHEDULE JCL statement

STARTBY specifies the preferred date and time that the job is selected for execution. JES2 attempts to position this job in the job queue so that the job is ready to be selected for execution at the specified time.

However, JES does not ensure that the job is selected at the specified time. The ability of the job to be selected depends on the system environment, system affinity, availability of initiators, and availability of resources, among other factors.

You specify a date and time when a job starts as shown in Example 4-37. In this example, the target date and time are 16:30, June 12, 2018.

Example 4-37 Sample JCL to specify STARTBY for specific date and time

```
//JOBX  JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//        REGION=0M
//SCHED SCHEDULE STARTBY=('16:30','2018/163')
//STEP01  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=SYS1.PARMLIB(JES3IN00),DISP=SHR
//SYSUT2   DD DUMMY
//SYSIN    DD DUMMY
/*
```

Another option is to specify how long a job waits for selection, as shown in Example 4-38. The target time the job should be selected is after 30 minutes from job submission in this case.

Example 4-38 Sample JCL to specify STARTBY for specific period

```
//JOBX  JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//        REGION=0M
//SCHED SCHEDULE STARTBY='+00:30'
//STEP01  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=SYS1.PARMLIB(JES3IN00),DISP=SHR
//SYSUT2   DD DUMMY
//SYSIN    DD DUMMY
/*
```

This function is not enabled by default. It must be enabled by using the following command:

```
$TJOBCLASS(c),PROMO_RATE=nn
```

Where *nn* specifies how many positions a job can be moved up the execution queue in one STARTBY aging cycle (1 minute=fixed value). The default value PROMO_RATE=0 means that the STARTBY function is disabled for the job class. You can also set this value in the PROMO_RATE parameter on JOBCLASS initialization statement.

4.9.3 WITH= on SCHEDULE JCL statement

In a JES3 environment, a user can control which system selects a job for execution that is based on JES3 set-up condition, without specifying the execution system. For example, if a volume is online only from one system, the job that requires the volume is automatically routed to the system.

In JES2, although the condition is not volume-online status, you can control which system can select a job for execution that is based on a specific job reference.

Use the WITH parameter to specify that the job must be run on the same system where another reference job is active. If the WITH parameter is used, the job is not eligible for execution until the reference job is active. In addition, the job can be run only on the same system where the reference job is active.

Jobs having a WITH specification can be submitted before or after the reference job is started or submitted. However, it is a best practice to submit a job after the reference job is started because the reverse sequence causes extra processor overhead.

The sample JCL that is shown in Example 4-39 specifies that JOBB must be run in the system where JOBA is running.

Example 4-39 Sample JCL to specify WITH parameter

```
//JOBB  JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//        REGION=0M
//SCHED SCHEDULE WITH=JOBA
//STEP01  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=SYS1.PARMLIB(JES3IN00),DISP=SHR
//SYSUT2   DD DUMMY
//SYSIN    DD DUMMY
/*
```

If the referenced job (JOBA) is not yet active, the referencing job (JOBB) waits for execution, as shown in Example 4-40. When JOBA starts, JOBB automatically starts on the system JOBA is running.

Example 4-40 Sample display for the job waiting for reference job active

```
$DJ9543
$HASP890 JOB(JOBB) 605
$HASP890 JOB(JOBB)      STATUS=(AWAITING EXECUTION),CLASS=A,
$HASP890                PRIORITY=8,SYSAFF=(ANY),HOLD=(NONE),
$HASP890                WITH=JOBA
```

4.10 SPOOL management

JES3 includes a spool partitioning function mainly to isolate certain types of work in a specific partition. This function can improve spool recovery by keeping critical spool data separate from noncritical data.

In a JES3 environment, it is easy to implement this functionality by using only JES3 `Inish` deck definitions, as shown in Example 4-41, Example 4-42, and Example 4-43. Only related parameters are described in these examples. The “SPECIAL” spool partition, which consists of SPOOL4, is reserved for CLASS=B and MSGCLASS=Y:

- Define spool partitions, as shown in Example 4-41.

Example 4-41 Define spool partition

```
SPART,NAME=NORMAL,DEF=YES
SPART,NAME=SPECIAL
```

- Assign each spool space to a defined spool partition, as shown in Example 4-42.

Example 4-42 Assign each spool space to a defined spool partition

```
TRACK,DDNAME=SPOOL1,SPART=NORMAL
TRACK,DDNAME=SPOOL2,SPART=NORMAL
TRACK,DDNAME=SPOOL3,SPART=NORMAL
TRACK,DDNAME=SPOOL4,SPART=SPECIAL
```

- Relate certain work to a specific spool partition, as shown in Example 4-43.

Example 4-43 Relate certain work to a specific spool partition

```
CLASS,NAME=A,SPART=NORMAL
CLASS,NAME=B,SPART=SPECIAL
SYSOUT,CLASS=X,SPART=NORMAL
SYSOUT,CLASS=Y,SPART=SPECIAL
```

However, JES2 does not have the same capabilities as JES3. Therefore, you must manage JES2 spool with some combination of JES2 functions.

This section describes some functions to help JES2 spool management.

SPOOL fencing

Standard JES2 processing allows all jobs to allocate track groups on all available spool volumes. Spool partitioning is a facility that is provided within JES2 that permits the specific identification of spool volumes from which a particular job or job class can allocate track groups. This facility is also referred to as spool fencing.

JES2 fences a job to an installation-defined number of volumes. In this form of fencing, the job starts with a zero spool partitioning mask work area. As the job allocates spool space, each volume that is used corresponds to a bit set in the mask. The job is forced to use volumes listed in the mask only. Minimum fencing is defined as setting the volume limit to “1”.

You also can implement SPOOL partitioning based on, for example, JOBCLASSes or JOBNAMEs, similar to JES3. The SPOOLDEF initialization statements and two installation exits, Exit 11 and Exit 12, provide methods for accessing and setting the spool partitioning mask work area.

For more information about the sample EXIT 11 and EXIT 12 to implement spool partitioning similar to JES3, see Appendix E, “SPOOL partitioning exits sample code” on page 233.

It is a best practice to use JES2 standard functions to manage JES2 spools instead of JES3 spool partitioning simulation by JES2 EXITs. This practice ensures future maintainability.

SPOOL affinity

JES2 also processes your fencing requirements based on the system affinity to those volumes.

Each spool volume includes masks of systems that can allocate space on that volume. Jobs are limited to the spool volumes associated with a system. You assign spool volumes to particular systems by using the **$T SPOOL** command, as shown in Example 4-44. (No initialization options are available to perform this task.)

Example 4-44 SYSLOG sample to modify spool affinity

```
$TSPOOL(BH5SP1),SYSAFF=SC74
$HASP893 VOLUME(BH5SP1) 211
$HASP893 VOLUME(BH5SP1)  STATUS=ACTIVE,DSNAME=SYS1.HASPACE,
$HASP893                 SYSAFF=(SC74),TGNUM=9975,TGINUSE=4414,
$HASP893                 TRKPERTGB=5,PERCENT=44,RESERVED=NO,
$HASP893                 MAPTARGET=NO
$HASP646 21.8981 PERCENT SPOOL UTILIZATION
```

Privileged space

This function is described in 1.3.2, “JES2 resiliency” on page 6.

Dynamic add/delete/allocate spool

You do not need to preallocate a spool data set. You can dynamically add or allocate spools in emergency situations.

SPOOL merge function

You can move data off one spool volume to another volume dynamically in emergency situations.

Dynamic expand of a spool (or checkpoint) data set

You can dynamically expand a spool or checkpoint data set into adjacent space on volumes for emergency recovery.

Reserved volumes

You can allocate JES2 spool volumes with the option RESERVED=Yes, which marks the spool volume as reserved for special processing and no new allocations are allowed. The RESERVED=No option clears the reserved attribute.

4.11 JES2 Disk Reader

Following the direction to support the main JES3 functions, JES2 includes a function that is similar to the JES3 Disk Reader that supports copying a member from a predefined concatenation of data sets to an internal reader, which passes the records in the member to JES2 input processing.

This function allows JCL to be submitted to JES2 from a concatenation without having to log on to TSO, submit a job, or run a started task.

To support this functionality on JES2, a logical concatenation of PDSs, PDSEs, and z/OS UNIX directories is used as source of members. Commands also are available that can be used to read the member and copy it to the internal reader.

Consider the following points:

- A concatenation SUBMITLIB is the source of the members.
- A statement SUBMITRDR defines defaults for the input device.
- A command $SUBMIT reads the members and passes them to JES2 INPUT processing.

4.11.1 SUBMITLIB concatenation

The SUBMITLIB concatenation is created by using the common code that is used to implement the PROCLIB concatenation that is in use by JES2. This concatenation allows for the creation of different lists of concatenated data sets or UNIX directories where the JES2 can locate JCL members then copy them to be processed by JES2 input processing (see Figure 4-2).

```
SUBMITLIB(ddname)    DD(n)=(DSName=name,[VOLser=volume,UNIT=unit])
                     DD(n)=(PATH=pathname)
                     CONDitional|UNCONDitional
```

Figure 4-2 SUBMITLIB statement syntax:

These concatenated lists can be defined by the JES2 initialization data set deck or added dynamically by using the **$ADD SUBMITLIB** command.

Also, the **$ADD**, **$DEL**, **$T**, and **$D** commands are available on JES2 to manage SUBMITLIB concatenation.

Adding a SUBMITLIB concatenation

The **$ADD SUBMITLIB** command can be used to dynamically define a new SUBMITLIB data set concatenation to be used by JES2 when submitting batch jobs by using the **$SUBMIT** command. The concatenations can contain any combination of partitioned data sets (PDSs) or file system paths.

If a file system path is specified, files in the path must be with 1 - 8 character file names that conform to standard PDS member names and can be accessed by using the FILEDATA=TEXT allocation option (see Example 4-45).

Example 4-45 Sample of $ADD SUBMITLIB command used to define a new SUBMITLIB to JES2

```
$ADD SUBMITLIB(SUBLIB00),DD(01)=(DSNAME=SYS1.JES2.SUBMTLIB)
IEF196I IEF237I 9788 ALLOCATED TO $SB00006
$HASP736 SUBMITLIB(SUBLIB00) 393
```

```
$HASP736 SUBMITLIB(SUBLIB00)
$HASP736                     DD(1)=(DSNAME=SYS1.JES2.SUBMTLIB,
$HASP736                     VOLSER=BH5CAT)
```

Modifying a SUBMITLIB concatenation

The **$T SUBMITLIB** command is used to modify a SUBMITLIB concatenation. By using this command, new data sets can be added to a concatenation, or existing data sets can be updated or deleted from the concatenation.

If a **$T SUBMITLIB** command is entered with no operands, concatenation is reallocated. This feature is useful to retry a failed allocation or to pick up updates to an existing data set (for example, a data set cataloged on a new volume or a data set is expanding into new extents).

Unlike PROCLIB processing, JES2 opens and closes the SUBMITLIB concatenations for every **$SUBMIT** command. This ability reduces the risk of errors when data sets in the concatenation are compressed (see Example 4-46).

Example 4-46 Sample of $T SUBMIT command used to add a PATH to an existent concatenation

```
$T SUBMITLIB(SUBLIB00),DD(02)=(PATH='/u/jes2/submitlib/sublib00')
IEF196I IEF237I 9788 ALLOCATED TO $SB00010
IEF196I IGD103I SMS UNIX FILE ALLOCATED TO DDNAME SYS00006
IEF196I IEF285I   SYS1.JES2.SUBMTLIB                            KEPT
IEF196I IEF285I   VOL SER NOS= BH5CAT.
$HASP736 SUBMITLIB(SUBLIB00) 426
$HASP736 SUBMITLIB(SUBLIB00)
$HASP736                     DD(1)=(DSNAME=SYS1.JES2.SUBMTLIB,
$HASP736                     VOLSER=BH5CAT),
$HASP736                     DD(2)=(PATH=/u/jes2/submitlib/subli
$HASP736                     b00)
```

Displaying a SUBMITLIB concatenation

The **$D SUBMITLIB** command is used to display SUBMITLIB concatenations that are defined to JES2 to be used when a **$SUBMIT** command is issued to submit a job (see Example 4-47).

Example 4-47 Sample of $D SUBMITLIB command used to display a SUBMITLIB concatenation

```
$D SUBMITLIB(SUBLIB00)
$HASP736 SUBMITLIB(SUBLIB00) 430
$HASP736 SUBMITLIB(SUBLIB00)
$HASP736                     DD(1)=(DSNAME=SYS1.JES2.SUBMTLIB,
$HASP736                     VOLSER=BH5CAT),
$HASP736                     DD(2)=(PATH=/u/jes2/submitlib/subli
$HASP736                     b00)
```

Deleting a SUBMITLIB concatenation

The **$DEL SUBMITLIB** command is used to remove a SUBMITLIB data set concatenation that is no longer needed. After the use count for the specified SUBMITLIB goes to zero, the data sets in the concatenation are unallocated (see Example 4-48).

Example 4-48 Sample of $DEL SUBMITLIB command used to remove a SUBMITLIB concatenation

```
$DEL SUBMITLIB(SUBLIB00)
IEF196I IEF285I   SYS1.JES2.SUBMTLIB                            KEPT
IEF196I IEF285I   VOL SER NOS= BH5CAT.
```

```
IEF196I IGD104I UNIX FILE WAS RETAINED, DDNAME IS (        )
IEF196I FILENAME IS (/u/jes2/submitlib/sublib00)
$HASP736 SUBMITLIB(SUBLIB00) 468
$HASP736 SUBMITLIB(SUBLIB00)
$HASP736                     DD(1)=(DSNAME=SYS1.JES2.SUBMTLIB,
$HASP736                     VOLSER=BH5CAT),
$HASP736                     DD(2)=(PATH=/u/jes2/submitlib/subli
$HASP736                     b00) - ELEMENT DELETED
```

4.11.2 SUBMITRDR initialization statement

The SUBMITRDR statement is similar to the INTRDR statement and is used to define the characteristics of the reader that used to read the input members from SUBMITLIB concatenation and then pass this member to JES2 input processing.

The DD_DEFAULT subparameter specifies the default SUBMITLIB concatenation that is used by the **$SUBMIT** command (see Figure 4-3).

```
SUBMITRDR AUTH=(DEVICE=YES|NO,JOB=YES|NO,SYSTEM=YES|NO)
      CLASS=jobclass,DD_DEFAULT=ddname,HOLD=YES|NO,
      PRTYINC=nn,PRTYLIM=nn,SYSAFF=(affinity_list),
      TRACE=YES|NO
```

Figure 4-3 SUBMITRDR statement syntax

4.11.3 $SUBMIT command

The use of the **$SUBMIT** command copies a member of a SUBMITLIB data set to the SUBMIT reader. The specified member of the SUBMITLIB data set is read and passed to JES2 input processing. Any valid record that can be passed to JES2 by using an internal reader can be passed to input processing by using the **$SUBMIT** command.

Default attributes for the submit reader can be specified on the SUBMITRDR statement.

Security for data that is passed over the submit reader is based on the issuer of the **$SUBMIT** command. If a user identity exists that is associated with the issuer of the command, that user is considered the submitter of the data stream. Commands and jobs are processed as if the originator of the command issued the commands or submitted the jobs.

Only one **$SUBMIT** command can be active on a member at a time. A second **$SUBMIT** command, while the first is still active, fails with a $HASP149 message, which indicates that a previous command is still active.

The existence of the default SUBMITLIB DD name is checked at the time that the **$SUBMIT** command is sued. If it does not exist, the **$SUBMIT** command fails with a $HASP665 message, which indicates that the SUBMITLIB was not found.

The use of the **$SUBMIT** command queues the processing to a subtask in the JES2 address space. The processing occurs asynchronous to the JES2 command processor. Error messages that are related to the member not being found or any other errors occur after the command processing completes.

Error messages that are associated with the submit command (such as the member not being found or an error reading the member) are routed back to the originating console.

Normal messages that are created by input phase processing are routed by the route code, as usual (see Figure 4-4).

```
$SUBMIT DDname=name,MEMBER=member,HOLD=YES|NO
```

Figure 4-4 $SUBMIT command syntax

4.12 JES2 Multi-Job NJE Stream

JES2 lifted the restriction for a single job object in an NJE job stream. Jes2 can send a set of jobs to a node and have them processed by input processing in the correct order. This support is based on a JOBGROUP defining jobs and all the jobs that are related to the job group in one NJE job stream.

This property is enforced on existing nodes that are receiving jobs from a JES3 migrated environment that support multiple jobs that are received from an NJE node and JES2 not. This support removes the restriction of only one job allowed by NJE submission.

This feature is supported on JES2 base code and does not require any external change on current JES2 environment.

If you want to run the jobs in destination node following a specific order, you can create a package by using the JOBGROUP definition and it is grouped JOBs into one NJE transmission unit. With the definition that is shown in Example 4-49, the JOBGROUP is processed first, and therefore exists on the receiving node before the grouped JOBs are processed.

In summary, packaging all JOBs and JOBGROUPs in a single /*XMIT unit is easier to understand and can eliminate potential NJE transmission timing issues.

Example 4-49 Sample of job using Multi-Job NJE Stream with JOBGROUP definition

```
//JOBXMITG JOB (),'ITSO REDBOOKS',CLASS=B,MSGCLASS=X,
//         MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID
//*
/*XMIT WTSCPLX7 DLM=ENDXMIT
//*
//*------------------------------------------------------------
//* JOBGROUP XMJG0001 : JOBXMITA --> JOBXMITB
//*------------------------------------------------------------
//XMJG0001 JOBGROUP
//*
//JOBXMITB GJOB
//         AFTER NAME=JOBXMITA,WHEN=(RC=0)
//*
//JOBXMITA GJOB
//*
//XMJG0001 ENDGROUP
//*------- JOB : JOBXMITA -------------------------------------
//JOBXMITA JOB (),'ITSO REDBOOKS',CLASS=B,MSGCLASS=X,
//         MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID
//*
//         SCHEDULE JOBGROUP=XMJG0001
//STEP1    EXEC PGM=IEFBR14
//*------- JOB : JOBXMITB -------------------------------------
```

```
//JOBXMITB JOB (),'ITSO REDBOOKS',CLASS=B,MSGCLASS=X,
//         MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID
//*
//         SCHEDULE JOBGROUP=XMJG0001
//STEP1    EXEC PGM=IEFBR14
//*-------------------------------------------------------------
ENDXMIT
```

5

JES procedure and initialization decks

This chapter describes the differences between the JES3 and JES2 initialization statements. As expected, many of the concepts are similar. However, the way that they are defined to the respective JES is likely to be different. Also, some functions or resources that are in one JES do not exist in the other JES.

Studying your JES2 initialization statements and mapping them to their JES3 equivalent (where possible) provides you with a valuable insight to how many changes you must make when migrating from JES3 to JES2.

The started procedures for JES2 and JES3 are also described.

This chapter includes the following topics:

5.1 Introduction

A JES2 JESplex consists of two or more z/OS systems that are running JES2 in the same Sysplex and all sharing spool and checkpoint data sets. JES2 uses the JES cross-system coupling services (JES XCF) for communicating JES2 member status and other data among the JES2 XCF group members in a multi-access spool (MAS) configuration.

Each JES2 member can read jobs from local and remote card readers, select jobs for processing, print, and punch results on local and remote output devices, and communicate with the operator. Each JES2 member in a Sysplex operates independently of the other JES2 members.

The JES2 members share a common job queue and a common output queue, which are on the checkpoint data sets. These common queues enable each JES2 member to share in processing the installation's workload. Jobs can run on whatever system is available and print or punch output on whatever system has an available device with the proper requirements.

A JES3 JESplex consists of a Global system and zero or more Local systems. The Global system is the first system to perform a cold or warm start following a JESplex-wide IPL.

During a cold or warm start, the Global system reads the initialization deck. When initialization of the Global is complete, any other systems in the JESplex might start JES3 as locals.

JES3 on a Local system communicates with the Global through XCF. A Local system never accesses the JES3 initialization deck. Instead, it obtains the information that it needs about the configuration by reading the checkpoint data set. It uses the information in the checkpoint to start communication with the Global system.

The JES3 Global function can be moved to a Local system during a planned or unplanned outage by performing a Dynamic System Interchange (DSI). The initialization deck is not read during a DSI. Instead, JES3 uses its checkpoint data set to bring the JESplex back to normal function.

5.2 Initialization statements

JES2 and JES3 require an initialization data set with all system definitions and characteristics that are related to the environment and how the JES works. This data set is read during the JES initialization process according to the following rules:

- For JES2, all members in a JESplex environment read the initialization data set during the initialization process.
- For JES3, only the Global system reads the initialization data set in a JESplex environment because the Local systems read the required information from the checkpoint.

The following examples include initialization parameters that are used by JES3 and the equivalent that is used during JES2 initialization. Based on these examples, a new JES2 environment can be built that is based on previous JES3 definitions. The statements were coded based on a current JES3 environment and tested by using the JES2 initialization deck checker process.

The JES3 definition for SPOOL and system-related data sets is shown in Example 5-1.

Example 5-1 JES3 system-related data sets definitions

```
/*
************************************************************************
* JES3 SYSTEM DATA SETS (SPOOL, JES3OUT, JES3JCT AND JES3DRDS)         *
************************************************************************
DYNALLOC,DDN=JES3JCT,DSN=SYS1.JCTG01
DYNALLOC,DDN=JES3OUT,DSN=SYS1.JES3OUT
DYNALLOC,DDN=JES3DRDS,DSN=SYS1.JES3DRDS
DYNALLOC,DDN=SPOOL1,DSN=SYS1.SPL001
DYNALLOC,DDN=SPOOL2,DSN=SYS1.SPL002
DYNALLOC,DDN=SPOOL3,DSN=SYS1.SPL003
*
BUFFER,BUFSIZE=2036,PAGES=(1024,512),GRPSZ=84,                         X
MINBUF=48,SPLIM=(10,20),TRUNC=YES
TRACK,DDNAME=SPOOL1,STT=(10,11),SPART=SPOOL1S
TRACK,DDNAME=SPOOL2,STT=(10,11),SPART=SPOOL1S
TRACK,DDNAME=SPOOL3,STT=(10,11),SPART=SPOOL2S
*
SPART,NAME=SPOOL1S,DEF=YES
SPART,NAME=SPOOL2S
*/
/**********************************************************************/
/* JES2 SYSTEM DATASETS (SPOOL AND CHECKPOINT)                        */
/**********************************************************************/
SPOOLDEF BUFSIZE=2036,DSNAME=SYS1.HASPACE,TGNUM=97728,VOLUME=JESSP,
         TGSIZE=84,LARGEDS=ALLOWED,SPOOLNUM=32,FENCE=YES,
         TGBPERVL=5,TGSIZE=45,TGSPACE=(MAX=260608,WARN=80),
         TRKCELL=12

CKPTDEF CKPT1=(STR=JES2_CKPT1,INUSE=YES),
        CKPT2=(STR=JES2_CKPT2,INUSE=YES),
        NEWCKPT1=(DSN=SYS1.JESCKPT1,VOL=JESSP1),
        NEWCKPT2=(DSN=SYS1.JESCKPT2,VOL=JESSP2)
```

JES3 procedure libraries concatenation definitions are shown in Example 5-2.

Example 5-2 JES statements to define PROCLIB concatenation

```
/*
************************************************************************
* JES3 PROCEDURE LIBRARIES :                                               *
************************************************************************
DYNALLOC,DDN=IATPLBST,DSN=SYS1.PROCLIB
DYNALLOC,DDN=IATPLBST,DSN=SYS1.IBM.PROCLIB
DYNALLOC,DDN=IATPLB01,DSN=SYS1.PROCLIB
DYNALLOC,DDN=IATPLB01,DSN=SYS1.TSO.PROCLIB
DYNALLOC,DDN=IATPLB01,DSN=SYS1.IBM.PROCLIB
*/
/**********************************************************************/
/* JES2 PROCEDURE LIBRARY DEFINITION                                  */
/**********************************************************************/
PROCLIB(PROC00)  DD(1)=(DSN=SYS1.PROCLIB),
                 DD(2)=(DSN=SYS1.IBM.PROCLIB),
                 UNCONDITIONAL
PROCLIB(PROC01)  DD(1)=(DSN=SYS1.PROCLIB),
                 DD(2)=(DSN=SYS1.TSO.PROCLIB),
                 DD(3)=(DSN=USER.PROCLIB),
                 UNCONDITIONAL
```

JSAM JES3 definitions and the equivalent JES2 definitions are shown in Example 5-3.

Example 5-3 JES3 JSAM definitions and equivalent JES2 statements

```
/*
***********************************************************************
* JES3 JSAM PARAMETERS                                                *
***********************************************************************
OPTIONS,DUMP=PRDMP,WANTDUMP=YES,JOBNO=(1,9999,9999),SE=10,MT=ON,       X
DUMPLINS=65535,INTRDR=20,XCFGRPNM=JESXCFG,DUPLOGON=YES,JOBTRACK=SYSPLEX
*
ENDJSAM
*/
/*********************************************************************/
/* JES2 DEFINITIONS                                                  */
/*********************************************************************/
INTRDR HOLD=NO,AUTH=(JOB=YES),RDINUM=20
JOBDEF JOBNUM=9999
MASDEF DORMANCY=(0,100),SHARED=CHECK,SYNCTOL=120,HOLD=0,
       AUTOEMEM=ON,RESTART=YES,CKPTLOCK=ACTION,LOCKOUT=1000,
       CYCLEMGT=AUTO,XCFGRPNM=JESXCFG,ENFSCOPE=SYSPLEX
```

JES3 standards definitions and the JES2 statements that are used to cover most of these standards are shown in Example 5-4. For some standards, no JES2 equivalent statement is available.

Example 5-4 JES3 standards definitions and the JES2 corresponding statements

```
/*
***********************************************************************
* STANDARDS FOR JES3                                                  *
***********************************************************************
STANDARDS,CICNT=(10,4),LINES=(150000,W),PRTY=6,SETUP=THWS,CARDS=(200), X
STCPMID=02,TSOPMID=03,TSOPROC=01,BYTES=(999999,W),PAGES=(1000,W),      X
FAILURE=CANCEL,MAXJOBST=3200,THWSSEP=PREFER
*
***********************************************************************
* Z/OS CONVERTER PARAMETERS                                           *
***********************************************************************
CIPARM,PARM=(40600300050031E00011X),PARMID=01,REGION=5M
CIPARM,PARM=(40600600050031E00011X),PARMID=02,REGION=5M
CIPARM,PARM=(40600300050031E00011Z),PARMID=03,REGION=5M
*/
/*********************************************************************/
/* JES2 DEFINITIONS FOR STANDARDS AND CONVERTER PARAMETERS           */
/*********************************************************************/
ESTLNCT NUM=150,INT=100000,OPT=0
ESTPUN  NUM=200,INT=10,OPT=0
ESTPAGE NUM=1000,INT=1000,OPT=0
ESTBYTE NUM=1000,INT=100,OPT=0
```

```
JOBCLASS(*) BLP=YES,COMMAND=IGNORE,JOURNAL=NO,MSGLEVEL=(1,1),
     PROCLIB=00,REGION=5M,SWA=ABOVE,TIME=(30,0)

JOBCLASS(STC) BLP=YES,COMMAND=DISPLAY,MSGCLASS=X,MSGLEVEL=(1,1),
     PROCLIB=00,REGION=5M,SWA=ABOVE,TIME=(60,0)

JOBCLASS(TSU) BLP=YES,COMMAND=DISPLAY,LOG=NO,MSGCLASS=Z,MSGLEVEL=(1,1),
     PROCLIB=01,REGION=5M,SWA=ABOVE,TIME=(30,0)

JOBDEF JOBNUM=32767,PRTYHIGH=14,PRTYJECL=YES,PRTYJOB=YES,PRTYLOW=6,
       PRTYRATE=48,RANGE=1-32767
```

The main processors definition in the JES3 environment with the equivalent JES2 member definition to identify the JES2 MAS members are shown in Example 5-5.

Example 5-5 JES3 Main processors definition and JES2 Member

```
/*
************************************************************************
* DEFINE JES3 MAIN PROCESSORS                                          *
************************************************************************
DEVICE,DTYPE=SYSMAIN,JNAME=SC74,                                        X
JUNIT=(,SC74,,ON,,SC75,,OFF)
DEVICE,DTYPE=SYSMAIN,JNAME=SC75,                                        X
JUNIT=(,SC75,,ON,,SC74,,OFF)
*
MAINPROC,NAME=SC74,SYSTEM=JES3,SELECT=SEL74
*
MAINPROC,NAME=SC75,SYSTEM=JES3,SELECT=SEL75
*/
/**********************************************************************/
/* JES2 DEFINITIONS FOR MAIN PROCESSORS - ADDED TO MASDEF KEYWORD     */
/**********************************************************************/
MEMBER(1)  NAME=SC75
MEMBER(2)  NAME=SC74
```

JES3 definitions to specify the job classes and the job class groups with the equivalent JES2 Jobclass definition are shown in Example 5-6.

Example 5-6 JES3 job execution classes and groups with the JES2 definition to job classes

```
/*
************************************************************************
* SELECT, GROUP AND CLASS DEFINITION                                   *
************************************************************************
*
SELECT,NAME=SEL74,                                                      X
SBAR=10,SAGER=03,SAGEL=14,LSTOR=32000,                                  X
GROUP=(GRA,10)
*
SELECT,NAME=SEL75,                                                      X
SBAR=10,SAGER=03,SAGEL=14,                                              X
GROUP=(GRB,30)
*
SELECT,NAME=DUMMY
```

```
*
GROUP,NAME=GRA,                                                        X
EXRESC=(*ALL,30,,,MANUAL),                                             X
DEF=YES
GROUP,NAME=GRB,MODE=WLM,                                               X
EXRESC=(*ALL,30,,,MANUAL),                                             X
*
CLASS,NAME=A,DEF=YES,GROUP=GRA,SDEPTH=30,LSTRR=0
CLASS,NAME=I,GROUP=GRA,SDEPTH=1,LSTRR=0,TDEPTH=1
CLASS,NAME=N,GROUP=GRB,SDEPTH=30,LSTRR=0,SPART=SPOOL2S
CLASS,NAME=V,GROUP=GRB,SDEPTH=30,LSTRR=0,SPART=SPOOL2S
*
*/
/**********************************************************************/
/* JES2 JOBCLASS DEFINITION                                           */
/**********************************************************************/
JOBCLASS(*)   AUTH=ALL,XEQCOUNT=MAX=30,SWA=ABOVE,GROUP=GRA,
              COMMAND=DISPLAY
JOBCLASS(I)   AUTH=ALL,XEQCOUNT=MAX=1,SWA=ABOVE,GROUP=GRA,
              COMMAND=DISPLAY
JOBCLASS(N)   AUTH=ALL,XEQCOUNT=MAX=30,SWA=ABOVE,GROUP=GRB,
              COMMAND=DISPLAY,MODE=WLM
JOBCLASS(V)   AUTH=ALL,XEQCOUNT=MAX=30,SWA=ABOVE,GROUP=GRB,
              COMMAND=DISPLAY,MODE=WLM
```

JES3 output processing definitions and the equivalent JES2 definitions that use the OUTCLASS statements are shown in Example 5-7.

Example 5-7 JES3 sysout definitions and the equivalent JES2 output classes

```
/*
************************************************************************
* OUTPUT SERVICE DEFAULT AND STANDARDS                                 *
************************************************************************
OUTSERV,TRAIN=ANY,WC=(P),WS=(D,T,F,C,CL,U,P,L),                        X
FORMS=STD,CDSTOCK=STD,                                                 X
CB=N,OUTLIM=16777215,OUTSVFCT=5,EXTOSENUM=NO
*
************************************************************************
* JES3 SYSOUT DEFINITION                                               *
************************************************************************
SYSOUT,CLASS=A,OVFL=OFF
SYSOUT,CLASS=M,OVFL=OFF,HOLD=TSO,TYPE=(PRINT),SPART=SPOOL2S
SYSOUT,CLASS=X,COPIES=0,HOLD=EXTWTR
SYSOUT,CLASS=Y,OVFL=OFF
SYSOUT,CLASS=Z,OVFL=OFF,HOLD=EXTWTR
*/
/**********************************************************************/
/* JES2 SYSOUT DEFINITION                                             */
/**********************************************************************/
OUTDEF JOENUM=6000,DSLIMIT=10M,STDFORM=STD

OUTCLASS(A)
OUTCLASS(M) OUTDISP=HOLD
OUTCLASS(X) OUTDISP=HOLD
OUTCLASS(Y)
OUTCLASS(Z) OUTDISP=HOLD
```

JES3 and JES2 console definition and the command prefix are shown in Example 5-8.

Example 5-8 JES3 and JES2 definitions to console and command prefix

```
/*
***********************************************************************
* CONSOLE SERVICE STANDARDS :                                         *
***********************************************************************
CONSTD,SYN=($),GLOBMPF=NO,DLOG=ON
*/
/*********************************************************************/
/* JES2 CONSOLE DEFINITION                                           */
/*********************************************************************/
CONDEF CONCHAR=$,BUFNUM=200,CMDNUM=1000,SCOPE=SYSTEM
```

JES3 FSS printers definitions and the same JES2 definitions are shown in Example 5-9.

Example 5-9 JES3 FSS printers definition and the equivalent JES2 definitions

```
/*
***********************************************************************
* JES3 FSS DEFINITIONS                                                *
***********************************************************************
FSSDEF,TYPE=WTR,FSSNAME=FSSPRT1,PNAME=PSFPRT1,                         X
SYSTEM=(SC74,SC75)
FSSDEF,TYPE=WTR,FSSNAME=FSSPRT2,PNAME=PSFPRT2,                         X
SYSTEM=(SC74,SC75)
FSSDEF,TYPE=WTR,FSSNAME=FSSPRT3,PNAME=PSFPRT3,                         X
SYSTEM=(SC74,SC75)
*/
/*********************************************************************/
/* JES2 FSS PRINTER DEFINITION                                       */
/*********************************************************************/
FSSDEF(FSSPRT1) PROC=PSFPRT1
FSSDEF(FSSPRT2) PROC=PSFPRT2
FSSDEF(FSSPRT3) PROC=PSFPRT3
```

JES3 printer definitions and the corresponding JES2 initialization statements are shown in Example 5-10.

Example 5-10 JES3 printer definition and the JES2 equivalent

```
/*
***********************************************************************
* JES3 FSS PRINTER DEFINITIONS                                        *
***********************************************************************
* PRT1
DEVICE,DTYPE=PRTAFP1,DGROUP=PRTGRP,                                    X
JNAME=PRT1,MODE=FSS,FSSNAME=FSSPRT1,                                   X
FORMS=(YES,STD),PM=(LINE,PAGE),                                        X
HEADER=YES,LINELIM=999999,PAGELIM=999999,                              X
JUNIT=(,SC74,S2,OFF,,SC75,S2,OFF),PDEFAULT=(CHARS,FCB)
*/
/*********************************************************************/
```

```
/* JES2 PRINTER DEFINITION                                           */
/*********************************************************************/
PRINTDEF SEPPAGE=LOCAL=HALF,TRANS=NO,NIFCB=STD,NIUCS=GT10,LINECT=60

PRT(1) CLASS=A,FSS=FSSPRT1,MODE=FSS,PRESELCT=YES,
       START=NO,TRKCELL=YES,WS=(Q,R)
```

The statements that are used by JES3 to define remote printer RJP are called RMT10. The equivalent JES2 definition that is used for RJE processing is shown in Example 5-11.

Example 5-11 JES3 RJP and JES2 RJE statements

```
/*
***********************************************************************
* JES3 REMOTE PRINTER SNA/RJP DEFINITION                              *
***********************************************************************
CONSOLE,TYPE=RJP,JNAME=RMT10,DEST=NONE,LEVEL=10,LL=80
RJPWS,N=RMT10,RD=1,PR=2,PU=1,C=R,COMPACT=YES,PL=2,TRACE=ON
*
DEVICE,DTYPE=RMTPRINT,JNAME=RMT10PR1,XLATE=NO,CHNSIZE=1,
CARRIAGE=NO,TRAIN=NO,HEADER=NO,DGROUP=RJPPRT
DEVICE,DTYPE=RMTPRINT,JNAME=RMT10PR2,XLATE=NO,CHNSIZE=1,
CARRIAGE=NO,TRAIN=NO,HEADER=NO,DGROUP=RJPPRT
*/
/*********************************************************************/
/* JES2 REMOTE PRINTER DEFINITION FOR SNA/RJE                        */
/*********************************************************************/
RMT(10)  DEVTYPE=LUTYPE1,BUFSIZE=512,COMPACT=YES,COMPRESS=YES,
         CONS=YES,DISCINTV=0,LINE=10,NUMPRT=2,NUMPUN=1,NUMRDR=1

R(10).PR(1) CCTL=YES,CKPTLINE=66,CKPTPAGE=10,CLASS=1,START=NO,
         SEPDS=YES,PRWIDTH=255,SELECT=PRINT1,ROUTECDE=(LOCAL,R10),
         WS=(W,R,Q,PMD,LIM/T,C,P)

R(10).PR(2) CCTL=YES,CKPTLINE=66,CKPTPAGE=10,CLASS=1,START=NO,
         SEPDS=YES,PRWIDTH=255,SELECT=PRINT2,ROUTECDE=(LOCAL,R10),
         WS=(W,R,Q,PMD,LIM/T,C,P)

R(10).PU(1) SELECT=PUNCH1,LRECL=80,SEP=NO

R(10).RD(1)
```

The basic JES3 NJE definitions and the JES2 APPL definitions that are used by JES2 to configure an NJE environment are shown in Example 5-12.

Example 5-12 Basic JES3 and JES2 NJE definitions

```
/*
***********************************************************************
* JES3 BDT SNA/NJE NODE DEFINITION                                    *
***********************************************************************
*
NJERMT,NAME=SYSJES,HOME=YES
*/
/*********************************************************************/
/* JES2 LOCAL NODE DEFINITION                                        */
```

```
/*********************************************************************/
APPL(SC74NJE) NODE=1

NJEDEF   DELAY=300,HDRBUF=(LIMIT=100,WARN=80),
         JRNUM=1,JTNUM=1,SRNUM=7,STNUM=7,LINENUM=5,MAILMSG=YES,
         MAXHOP=0,NODENUM=999,OWNNODE=1,PATH=1,CONNECT=(YES,1),
         RESTMAX=0,RESTNODE=100,RESTTOL=0,TIMETOL=0

LOGON(1) APPLID=SC74NJE

NODE(1)  NAME=WTSCPLX7,PATHMGR=YES,SUBNET=MYJES
```

Other JES2 parameters that are required to complete the JES2 initialization configuration are shown in Example 5-13.

Example 5-13 JES2 extra parameters that are used by JES2 initialization

```
/*********************************************************************/
/* ADDITIONAL JES2 DEFINITIONS                                       */
/*********************************************************************/
/*********************************************************************/
/* TWS JES2 EXITS DEFINITIONS WITH TWO DIFFERENT WAY                */
/*********************************************************************/
EXIT(11) ENABLE,ROUTINES=(JES2X011)
LOAD(JES2X011)
EXIT(12) ROU=(JES2X012),STATUS=ENABLED
LOAD(JES2X012)
/*********************************************************************/
/* JES2 INITIATORS DEFINITION                                       */
/*********************************************************************/
INITDEF PARTNUM=30
INIT(01) CLASS=ABCDE,START=NO
INIT(02) CLASS=ABCDE,START=NO
INIT(03) CLASS=ABCDE,START=NO
INIT(04) CLASS=ABCDE,START=NO
INIT(05) CLASS=ABCDE,START=NO
INIT(06) CLASS=ABCDE,START=NO
INIT(07) CLASS=ABCDE,START=NO
INIT(08) CLASS=ABCDE,START=NO
INIT(09) CLASS=ABCDE,START=NO
INIT(10) CLASS=ABCDE,START=NO
INIT(11) CLASS=ABCDE,START=NO
INIT(12) CLASS=ABCDE,START=NO
INIT(13) CLASS=ABCDE,START=NO
INIT(14) CLASS=ABCDE,START=NO
INIT(15) CLASS=ABCDE,START=NO
INIT(16) CLASS=ABCDE,START=NO
INIT(17) CLASS=ABCDE,START=NO
INIT(18) CLASS=ABCDE,START=NO
INIT(19) CLASS=ABCDE,START=NO
INIT(20) CLASS=ABCDE,START=NO
INIT(21) CLASS=ABCDE,START=NO
INIT(22) CLASS=ABCDE,START=NO
INIT(23) CLASS=ABCDE,START=NO
INIT(24) CLASS=ABCDE,START=NO
INIT(25) CLASS=ABCDE,START=NO
INIT(26) CLASS=ABCDE,START=NO
INIT(27) CLASS=ABCDE,START=NO
INIT(28) CLASS=ABCDE,START=NO
INIT(29) CLASS=ABCDE,START=NO
INIT(30) CLASS=ABCDE,START=NO
/*********************************************************************/
/* JES2 TCP/IP AND SNA LINE DEFINITIONS                             */
/*********************************************************************/
NETSRV1 SOCKET=LOCAL,STACK=TCPIP
LINE(1)  UNIT=TCP
LINE(2)  UNIT=TCP
LINE(3)  UNIT=TCP
LINE(10) UNIT=SNA
SOCKET(WTSCPLX7) NODE=1
TPDEF  BELOWBUF=(SIZE=3960),EXTBUF=(SIZE=3840),SESSION=31
/*********************************************************************/
/* JES2 SPOOL OFFLOAD DEFINITION                                     */
/*********************************************************************/
OFFLOAD1 DSN=SYSU.&SYSNAME..OFFLOAD1,UNIT=(,1),LABEL=SL
                             /**** OFFLOAD JOB RECEIVER **************/
OFF(1).JR CLASS=,CREATOR=,HOLD=,JOBNAME=,
         MOD=(CLASS=,HOLD=,ROUTECDE=,SYSAFF=),
         NOTIFY=NO,ROUTECDE=(),START=NO,SYSAFF=(),
         WS=(CLASS/)

OFF(1).JT CLASS=,CREATOR=,DISP=DELETE,HOLD=,
         JOBNAME=,NOTIFY=NO,ROUTECDE=(),START=NO,SYSAFF=(),
         VOLUME=(),WS=(CLASS/)

OFF(1).SR BURST=,CREATOR=,FCB=,FLASH=,FORMS=,HOLD=,JOBNAME=,
         MOD=(BURST=,FCB=,FLASH=,FORMS=,HOLD=,OUTDISP=,PRMODE=,
              QUEUE=,ROUTECDE=,UCS=,WRITER=),
         NOTIFY=NO,PRMODE=(),QUEUE=,ROUTECDE=(),START=YES,UCS=,
         WRITER=,WS=(/)

OFF(1).ST BURST=,CREATOR=,DISP=DELETE,FCB=,FLASH=,FORMS=,HOLD=,
         JOBNAME=,LIMIT=(0-*),NOTIFY=NO,PLIM=(0-*),PRMODE=(),
         QUEUE=,ROUTECDE=(),START=YES,UCS=,VOLUME=(),WRITER=,WS=(/)
/*********************************************************************/
/* JES2 DISK READER DEFINITION                                       */
/*********************************************************************/
SUBMITLIB(SLTEST) DD(01)=(DSNAME=SYS1.JES2.SUBLIB.TEST),UNCONDITIONAL
SUBMITLIB(SLPROD) DD(01)=(DSNAME=SYS1.JES2.SUBLIB.PROD),UNCONDITIONAL

SUBMITRDR AUTH=(DEVICE=NO,JOB=NO,SYSTEM=NO),
          CLASS=A,DD_DEFAULT=SLTEST,HOLD=NO,
          PRTYINC=01,PRTYLIM=08,
          TRACE=NO
/*********************************************************************/
/* JES2 POLICY LIBRARY DEFINTIONS                                    */
/*********************************************************************/
PLCYLIB(PLCYLIB00),DD(01)=(DSNAME=SYS1.JES2.PLCYLIB),UNCONDITIONAL
```

JES3 initialization statements that do not have JES2 equivalent are shown in Example 5-14.

Example 5-14 JES3 initialization statements with no JES2 correspondence

```
/*
***********************************************************************
* RESIDENCY JES3 OPTIONS                                              *
***********************************************************************
RESCTLBK,FCT=128
***********************************************************************
* SPECIFIC DYNAMIC DATASET ALLOCATION AND RESERVED DATASET NAMES      *
***********************************************************************
DYNALDSN,BYPASS=(TEST.*,SYSA.*)
DYNALDSN,BYPASS=(PRD.*,PRDO.*)
DYNALDSN,BYPASS=(?.?.OUTLIST,?.?.LIST,?.PROFILE)
DYNALDSN,PROTECT=(*)
*
RESDSN,DSN=(SYN1.LINKLIB)
RESDSN,DSN=(SYS1.LPALIB)
RESDSN,DSN=(SYS1.MACLIB)
RESDSN,DSN=(SYS1.MIGLIB)
RESDSN,DSN=(SYS1.NUCLEUS)
RESDSN,DSN=(USER.PROCLIB)
***********************************************************************
* MAIN DEVICE SCHDULING (MDS)                                         *
***********************************************************************
SETPARAM,FETCH=NO,DSN=26,                                              X
SMSSETUP=NO,
MDSLOG=S1,REMOUNT=255
***********************************************************************
* SETNAMES FOR DEVICES                                                *
***********************************************************************
SETNAME,XTYPE=3390,NAMES=(DASD,3390)
***********************************************************************
* JES3 DASD DEVICES                                                   *
***********************************************************************
DEVICE,XTYPE=(3390,DA),XUNIT=(1000,*ALL,S7,OFF),NUMDEV=1024
DEVICE,XTYPE=(3390,DA),XUNIT=(2000,*ALL,S7,ON),NUMDEV=1024
*/
```

5.2.1 Verifying the JES initialization deck

JES2 and JES3 include an initialization stream checker that can be used to validate the initialization statements syntax and how the initialization parameters can affect the current JES environment.

The JES3 initialization stream checker is processed by the IATUTIS program that allows the system programmer to verify that the deck has no errors before a scheduled restart. The initialization stream checker also detects most syntax errors and some logical errors in the initialization stream.

Installations that still have disk or tape DEVICE statements might use option 2.4 in the HCD ISPF panels to create members that are then pointed to by the STG1CODE DD statement in the checker job. The initialization deck checker then verifies that the DEVICE statements agree with the devices in the HCD.

A sample JCL for initialization deck checking is shown in Example 5-15.

Example 5-15 Sample JCL for IATUTIS initialization deck checker

```
//INITCHK    JOB 'ACCTINFO','NAME',MSGLEVEL=(1,1),
//                MSGCLASS=R,...
//IATUTIS    EXEC PGM=IATUTIS,PARM='P=1F1R'
//STEPLIB    DD DSN=SYS1.SIATLIB,DISP=SHR
//JESABEND   DD DUMMY
//JES3IN     DD DSN=INIT.PARMLIB(JES3IN00),DISP=SHR
//JES3OUT    DD SYSOUT=*
//STG1CODE   DD DSN=INSTALL.JES3,DISP=SHR
//IATPLBST   DD DSN=SYS1.PROCLIB,DISP=SHR
//
```

As with the JES3 initialization stream checker, JES2 initialization data set checker allows installations to verify their initialization data sets without having to start a JES2 subsystem. The process can detect syntax errors in initialization statements and problems with settings that might prevent JES2 from starting. The checks can verify that the statements are valid for a cold start or a warm start.

If parameters are verified for a warm start, you must run the checker within a SYSPLEX with an MAS active member. The checker uses XCF messaging to extract information from the active MAS member to verify whether the parameters are valid on a warm start and to perform more analysis of resource usage. A sample JCL to run the JES2 initialization data set checker as a batch job is shown in Example 5-16.

Example 5-16 Sample JCL to run the JES2 initialization deck checker as a batch job

```
//INITJ2CK JOB (),’PROGRAMMER NAME’,CLASS=B,MSGCLASS=X,
// MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID
//*
//HASCHECK EXEC PGM=HASJESCK,PARM=’LIST’
//HASPLIST DD SYSOUT=*
//HASPPARM DD DISP=SHR,DSN=SYS1.PARMLIB(J2DFAULT)
//
```

HASPLIST output details

When the initialization data set checker is run, the normal JES2 initialization processing also is run, with all messages being captured in data set that is associated with the HASPLIST DD. The initialization statement listing is also placed in the HASPLIST DD that is based on the LIST and NOLIST PARM= value and initialization statement.

A sample output that is produced by JES2 initialization deck checker is shown in Example 5-17 on page 97, Example 5-18 on page 98, Example 5-19 on page 99, and Example 5-20 on page 99.

Example 5-17 Sample HASPLIST Parge1: Initialization parameters

```
1         JES2 parameter library listing   SYS1.PARMLIB(J2DFAULT)                                      2019.176  PAGE   1
-Reading SYS1.PARMLIB(J2DFAULT)
 PARMLIB      STMT      1                CONDEF CONCHAR=$,BUFNUM=200,CMDNUM=1000,DISPMAX=1000
 PARMLIB      STMT      2                SMFDEF BUFNUM=50
 PARMLIB      STMT      3                SPOOLDEF BUFSIZE=3856,DSNAME=SYS1.HASPACE,TGNUM=97728,VOLUME=BH5SP,TGSIZE=60,
                                         LARGEDS=ALLOWED
 PARMLIB      STMT      4                CKPTDEF  CKPT1=(STR=JES2CKPT_2,INUSE=YES),CKPT2=(DSN=SYS1.JES2.CKPT2,VOL=BH5JC2,
                                         INUSE=YES),NEWCKPT1=(DSN=SYS1.JES2.CKPT1,VOL=BH5JC1),NEWCKPT2=(STR=JES2CKPT_1),
                                         MODE=DUPLEX,DUPLEX=ON
 PARMLIB      STMT      5                MASDEF   DORMANCY=(0,100),SHARED=NOCHECK,SYNCTOL=120,HOLD=0,AUTOEMEM=ON,
                                         RESTART=YES,CKPTLOCK=ACTION,LOCKOUT=1000,CYCLEMGT=AUTO
 PARMLIB      STMT      6                MEMBER(1)  NAME=SC75
 PARMLIB      STMT      7                MEMBER(2)  NAME=SC74
 PARMLIB      STMT      8                PROCLIB(PROC00) DD01=(DSN=SYS1.PROCLIB),DD02=(DSN=SYS1.IBM.PROCLIB)
 PARMLIB      STMT      9                LOAD(JES2X011)
 PARMLIB      STMT     10                EXIT(11) ENABLE,ROU=(EXIT011)
 PARMLIB      STMT     11                LOAD(JES2X012) STOR=CSA
 PARMLIB      STMT     12                EXIT(12) ENABLE,ROU=(EXIT012)
 PARMLIB      STMT     13                JOBDEF JOBNUM=3000
 PARMLIB      STMT     14                OUTDEF JOENUM=6000
 PARMLIB      STMT     15                APPL(SC75NJE) NODE=1
 PARMLIB      STMT     16                APPL(SCHNJE)  NODE=3
 PARMLIB      STMT     17                NJEDEF   DELAY=300,HDRBUF=(LIMIT=100,WARN=80),JRNUM=1,          JTNUM=1,SRNUM=7,
                                                   STNUM=7,LINENUM=5,       MAILMSG=YES,MAXHOP=0,         NODENUM=999,
                                         OWNNODE=1,        PATH=1,RESTMAX=0,RESTNODE=100,     RESTTOL=0,TIMETOL=0,
                                          CONNECT=(YES,1)
 PARMLIB      STMT     18                LOGON(1) APPLID=SC75NJE
 PARMLIB      STMT     19                NODE(1)    NAME=WTSCPLX7,PATHMGR=YES,SUBNET=MYJES
 PARMLIB      STMT     20                NODE(2)    NAME=WTSCMXA,PATHMGR=YES,SUBNET=WTSCMXA
 PARMLIB      STMT     21                NODE(3)    NAME=WTSCNET,PATHMGR=YES,SUBNET=WTSCMXA
 PARMLIB      STMT     22                NODE(4)    NAME=WTSCPLX1,PATHMGR=YES,SUBNET=WTSCMXA
 PARMLIB      STMT     23                NODE(5)    NAME=WTSCPLX2,PATHMGR=YES,SUBNET=WTSCMXA
 PARMLIB      STMT     24                NODE(6)    NAME=WTSCPLX4,PATHMGR=NO,SUBNET=WTSCMXA
 PARMLIB      STMT     25                NODE(7)    NAME=WTSCPLX5,PATHMGR=YES,SUBNET=WTSCMXA
 PARMLIB      STMT     26                NODE(9)    NAME=WTSCPLX8,PATHMGR=YES,SUBNET=WTSCMXA
 PARMLIB      STMT     27                NODE(10)   NAME=WTSCPLX9,PATHMGR=NO,SUBNET=WTSCMXA
 PARMLIB      STMT     28                NODE(11)   NAME=WTSCPOK,PATHMGR=NO,SUBNET=WTSCMXA
 PARMLIB      STMT     29                NODE(12)   NAME=TRAINER,PATHMGR=YES,SUBNET=WTSCMXA
 PARMLIB      STMT     30                NODE(13)   NAME=WTSC58,PATHMGR=YES,SUBNET=WTSCMXA
 PARMLIB      STMT     31                NODE(14)   NAME=WTSC59,PATHMGR=YES,SUBNET=WTSCMXA
 PARMLIB      STMT     32                NODE(15)   NAME=WTSC60,PATHMGR=YES,SUBNET=WTSCMXA
 PARMLIB      STMT     33                NODE(16)   NAME=WTSC76,PATHMGR=YES,SUBNET=WTSCMXA
 PARMLIB      STMT     34                NODE(17)   NAME=WTSC90,PATHMGR=YES,SUBNET=WTSCMXA
 PARMLIB      STMT     35                NODE(18)   NAME=LABSERV,PATHMGR=YES,SUBNET=WTSCMXA
 PARMLIB      STMT     36                NODE(19)   NAME=DSTSC01,PATHMGR=YES,SUBNET=WTSCMXA
 PARMLIB      STMT     37                NODE(20)   NAME=DSTSC02,PATHMGR=YES,SUBNET=WTSCMXA
 PARMLIB      STMT     38                NODE(21)   NAME=PLPSC,PATHMGR=NO,SUBNET=WTSCMXA
 PARMLIB      STMT     39                NODE(22)   NAME=IBMUS,PATHMGR=NO,SUBNET=WTSCMXA
 PARMLIB      STMT     40                NODE(101)  NAME=SNJMAS3,PATHMGR=NO,SUBNET=WTSCMXA
 PARMLIB      STMT     41                TPDEF  BELOWBUF=(SIZE=3960),EXTBUF=(SIZE=3840),SESSION=31
 PARMLIB      STMT     42                JOBCLASS(TSU) AUTH=ALL,         BLP=YES,         LOG=NO,CONDPURG=NO,
                                         REGION=0M,       OUTPUT=YES,SWA=ABOVE,        MSGCLASS=S,TIME=(1440,0)
 PARMLIB      STMT     43                JOBCLASS(STC) AUTH=ALL,         BLP=NO,          LOG=NO,REGION=0M,
                                         CONDPURG=NO,     OUTPUT=YES,SWA=ABOVE,        MSGCLASS=S,TIME=(1440,0)
 PARMLIB      STMT     44                JOBCLASS(*)   AUTH=ALL,         BLP=YES,         LOG=YES,COMMAND=DISPLAY,
                                         MODE=JES,        CONDPURG=NO,JOURNAL=YES,      RESTART=YES,     XEQCOUNT=MAX=*,
                                         MSGLEVEL=(1,1),   REGION=2M,SWA=ABOVE,TIME=(450,00)
 PARMLIB      STMT     45                JOBCLASS(W)   AUTH=ALL,         BLP=YES,         LOG=YES,COMMAND=DISPLAY,
                                         MODE=WLM,        CONDPURG=NO,JOURNAL=YES,      RESTART=YES,     XEQCOUNT=MAX=*,
                                         MSGLEVEL=(1,1),   REGION=2M,SWA=ABOVE,TIME=(450,00)
 PARMLIB      STMT     46                JOBCLASS(L)   AUTH=ALL,         BLP=YES,         LOG=YES,COMMAND=DISPLAY,
                                         MODE=WLM,        CONDPURG=NO,JOURNAL=YES,      RESTART=YES,     XEQCOUNT=MAX=*,
                                         MSGLEVEL=(1,1),   REGION=2M,SWA=ABOVE,TIME=(450,00)
 PARMLIB      STMT     47                JOBCLASS(M)   AUTH=ALL,         BLP=YES,         LOG=YES,COMMAND=DISPLAY,
                                         MODE=WLM,        CONDPURG=NO,JOURNAL=YES,      RESTART=YES,     XEQCOUNT=MAX=*,
                                         MSGLEVEL=(1,1),   REGION=2M,SWA=ABOVE,TIME=(450,00)
 PARMLIB      STMT     48                JOBPRTY1 PRIORITY=9,TIME=1
 PARMLIB      STMT     49                JOBPRTY2 PRIORITY=8,TIME=2
 PARMLIB      STMT     50                JOBPRTY3 PRIORITY=7,TIME=4
 PARMLIB      STMT     51                JOBPRTY4 PRIORITY=6,TIME=8
 PARMLIB      STMT     52                JOBPRTY5 PRIORITY=5,TIME=16
 PARMLIB      STMT     53                JOBPRTY6 PRIORITY=4,TIME=32
 PARMLIB      STMT     54                JOBPRTY7 PRIORITY=3,TIME=64
 PARMLIB      STMT     55                JOBPRTY8 PRIORITY=2,TIME=128
 PARMLIB      STMT     56                JOBPRTY9 PRIORITY=1,TIME=256
 PARMLIB      STMT     57                ESTLNCT NUM=4,INT=4000
 PARMLIB      STMT     58                OUTCLASS(A)
 PARMLIB      STMT     59                OUTCLASS(B)   OUTPUT=PUNCH
 PARMLIB      STMT     60                OUTCLASS(C)
 PARMLIB      STMT     61                OUTCLASS(D)   OUTDISP=(HOLD,HOLD)
 PARMLIB      STMT     62                OUTCLASS(E-I)
 PARMLIB      STMT     63                OUTCLASS(J)   BLNKTRNC=NO
 PARMLIB      STMT     64                OUTCLASS(K)   OUTPUT=PUNCH
 PARMLIB      STMT     65                OUTCLASS(L)   OUTDISP=(HOLD,HOLD)
 PARMLIB      STMT     66                OUTCLASS(M-P)
 PARMLIB      STMT     67                OUTCLASS(Q)   OUTDISP=(HOLD,HOLD)
 PARMLIB      STMT     68                OUTCLASS(R)
 PARMLIB      STMT     69                OUTCLASS(S-T) OUTDISP=(HOLD,HOLD)
 PARMLIB      STMT     70                OUTCLASS(U-W)
 PARMLIB      STMT     71                OUTCLASS(X)   OUTDISP=(HOLD,HOLD)
 PARMLIB      STMT     72                OUTCLASS(Y)
 PARMLIB      STMT     73                OUTCLASS(Z)   OUTPUT=DUMMY,OUTDISP=(PURGE,PURGE),TRKCELL=NO
 PARMLIB      STMT     74                OUTCLASS(0-9)
 PARMLIB      STMT     75                INITDEF PARTNUM=50
 PARMLIB      STMT     76                I1 CLASS=ABCDE,START
 PARMLIB      STMT     77                I2 CLASS=ABCDE,START
 PARMLIB      STMT     78                I3 CLASS=ABCDE,START
 PARMLIB      STMT     79                I4 CLASS=ABCDE,START
 PARMLIB      STMT     80                I5 CLASS=ABCDE,START
 PARMLIB      STMT     81                I6 CLASS=ABCDE,START
 PARMLIB      STMT     82                I7 CLASS=ABCDE,START
 PARMLIB      STMT     83                I8 CLASS=ABCDE,START
 PARMLIB      STMT     84                I9 CLASS=ABCDE,START
 PARMLIB      STMT     85                I10 CLASS=ABCDE,START
 PARMLIB      STMT     86                I11 CLASS=ABCDE,DRAIN
 PARMLIB      STMT     87                I12 CLASS=ABCDE,DRAIN
 PARMLIB      STMT     88                I13 CLASS=ABCDE,DRAIN
 PARMLIB      STMT     89                I14 CLASS=ABCDE,DRAIN
 PARMLIB      STMT     90                I15 CLASS=ABCDE,DRAIN
 PARMLIB      STMT     91                I16 CLASS=ABCDE,DRAIN
 PARMLIB      STMT     92                I17 CLASS=ABCDE,DRAIN
 PARMLIB      STMT     93                I18 CLASS=ABCDE,DRAIN
 PARMLIB      STMT     94                I19 CLASS=ABCDE,DRAIN
 PARMLIB      STMT     95                I20 CLASS=ABCDE,DRAIN
 PARMLIB      STMT     96                I21 CLASS=ABCDE,DRAIN
 PARMLIB      STMT     97                I22 CLASS=ABCDE,DRAIN
 PARMLIB      STMT     98                I23 CLASS=ABCDE,DRAIN
 PARMLIB      STMT     99                I24 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    100                I25 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    101                I26 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    102                I27 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    103                I28 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    104                I29 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    105                I30 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    106                I31 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    107                I32 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    108                I33 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    109                I34 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    110                I35 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    111                I36 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    112                I37 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    113                I38 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    114                I39 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    115                I40 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    116                I41 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    117                I42 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    118                I43 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    119                I44 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    120                I45 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    121                I46 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    122                I47 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    123                I48 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    124                I49 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    125                I50 CLASS=ABCDE,DRAIN
 PARMLIB      STMT    126                FSSDEF(PSF1)     PROC=WTR3800S,HASPFSSM=FSSM3211
 PARMLIB      STMT    127                FSSDEF(PSF2)     PROC=WTR3800S,HASPFSSM=FSSM3211
 PARMLIB      STMT    128                FSSDEF(BLUEBERY) PROC=PWTR,HASPFSSM=FSSM3211
 PARMLIB      STMT    129                FSSDEF(RASPBERY) PROC=PWTRQ,HASPFSSM=FSSM3211
 PARMLIB      STMT    130                FSSDEF(LEMON)    PROC=PWTROLD,HASPFSSM=FSSM3211
 PARMLIB      STMT    131                FSSDEF(HFAM)     PROC=PWTRHGF,HASPFSSM=FSSM3211
 PARMLIB      STMT    132                INTRDR HOLD=NO,AUTH=(JOB=YES),RDINUM=4
 PARMLIB      STMT    133                PRINTDEF SEPPAGE=LOCAL=HALF,TRANS=NO,NIFCB=STD3,NIUCS=GT10,LINECT=60
 PARMLIB      STMT    134                LINE(1)  UNIT=TCP
 PARMLIB      STMT    135                LINE(2)  UNIT=TCP
 PARMLIB      STMT    136                LINE(3)  UNIT=TCP
 PARMLIB      STMT    137                LINE(4)  UNIT=TCP
 PARMLIB      STMT    138                LINE(5)  UNIT=TCP
 PARMLIB      STMT    139                LINE(6)  UNIT=TCP
 PARMLIB      STMT    140                LINE(7)  UNIT=TCP
 PARMLIB      STMT    141                LINE(8)  UNIT=TCP
 PARMLIB      STMT    142                LINE(9)  UNIT=TCP
 PARMLIB      STMT    143                NETSRV1 SOCKET=LOCAL,STACK=TCPIP
 PARMLIB      STMT    144                SOCKET(WTSCPLX7) NODE=1
 PARMLIB      STMT    145                SOCKET(WTSCMXA)  NODE=2,   IPADDR=10.12.6.131
 PARMLIB      STMT    146                INPUTDEF JES3JECL=PROCESS
 PARMLIB      STMT    147                JECLDEF JES3=(MAIN=PROCESS,FORMAT=PROCESS,NET=PROCESS,NETACCT=PROCESS)
 PARMLIB      STMT    148                SUBMITLIB(SLTEST) DD(01)=(DSNAME=LUTZ.SUBLIB.TEST),UNCONDITIONAL
 PARMLIB      STMT    149                SUBMITLIB(SLPROD) DD(01)=(DSNAME=LUTZ.SUBLIB.PROD),UNCONDITIONAL
 PARMLIB      STMT    150                SUBMITRDR AUTH=(DEVICE=NO,JOB=NO,SYSTEM=NO),CLASS=A,DD_DEFAULT=SLTEST,HOLD=NO,

PRTYINC=01,PRTYLIM=08,TRACE=NO
```

Example 5-18   Sample HASPLIST Pages 4 and 5: Initialization parameters and diagnostic report

```
1         JES2 parameter library listing                                                              2019.176  PAGE   5
-DIAGNOSTIC   INFO                       $HASP442 INITIALIZATION STATEMENTS CONFLICTING WITH SAVED VALUES FOLLOW:
 DIAGNOSTIC   WARNING                    $HASP496 OUTDEF JOENUM=6000 SAVED VALUE OF 10000 WILL BE USED
 DIAGNOSTIC   WARNING                    $HASP496 CKPTSPAC BERTNUM=4600 SAVED VALUE OF 2100 WILL BE USED
 DIAGNOSTIC   INFO                       $HASP537 THE CURRENT CHECKPOINT USES 722 4K
RECORDS
```

Example 5-19   Sample HASPLIST Pages 6 and 7: Statistics and Resource usage report

```
1         JES2 parameter library listing                                                               2019.176  PAGE   6
-
 Initialization data sets read:

          Data set name                                           VOLSER      Unit    Records
          ------------------------------------------------------  ------  --------  ---------
          SYS1.PARMLIB(J2DFAULT)                                  BH5CAT                  207
1         JES2 parameter library listing                                                               2019.176  PAGE   7
-
 Resource usage information:

 JQEs     TYPE        ACTIVE  COMPLETE       JOEs     TYPE         COUNT       TGs      TYPE           COUNT       INUSE
          -------- --------- ---------                -------- ---------                -------- ----------- -----------
          BATCH            1     1,134                WORK         4,022                DEFINED       40,167      20,305
          STC            107       335                CHAR            14                ACTIVE        40,017      20,395
          TSU              2        14                INDEX            0                FREE          19,622
          JOBGROUP         0        12                FREE         5,964
          INTERNAL        11
          FREE         1,384

 BERTs    TYPE         COUNT  CB COUNT       ZJCs     TYPE         COUNT       Jobnum   Description      Value
          -------- --------- ---------                -------- ---------                ------------ ---------
          INTERNAL        34         4                JOBGROUP         9                Low Range            1
          JQE            265       254                DEP JOB         28                High Range       9,999
          CAT            144        48                DEPENDNT        24                In Use           1,616
          WSCQ             8         2                FREE           939
          DJBQ             0         0
          JOE              6         6
          DAS              0         0
          GRP              2         2
          FREE         1,641

 Recommendations:

                            Current      Current Percent Usage per     Max with  Recommended
                              Limit        Usage   Usage   JQE/JOE  max JQE/JOE    min limit
                        -----------  ----------- ------   --------  -----------  -----------
          JQEs                3,000        1,616   53.86                               3,000
          Job Numbers         9,999        1,616   16.16      1.00        3,000        3,000
          JOEs               10,000        4,036   40.36      2.69        8,070        9,000
          Active TGs         40,017       20,395   50.96     12.56       37,680       40,000
          BERTs               2,100          459   21.85                               2,000
           JQE BERTs                         265              0.16          480
           JOE BERTs                           6              0.00            0
```

Example 5-20   Sample HASPLIST Page 8: Summary report

```
1         JES2 parameter library listing                                                               2019.176  PAGE   8
-
 Summary report:

          Member name                 SC74
          NJE node name               WTSCPLX7
          JESXCF group name           WTSCPLX7
          MVS system name             SC74
          MVS SYSPLEX name            PLEX75
          Checkpoint data             Obtained
          Checker version             z/OS 2.4

 Error Summary:

          Type                          Count
          --------------------------  -------
          Warnings                          2
          Init statement errors             0
          Validation errors                 0
          Read/OPEN errors                  0
          Configuration errors              0
          Exit requested termination        0
          --------------------------  -------
          Total error count
2
```

After the initialization statements are processed, the processing attempts to access the runtime data. If the runtime data is available, the normal verification processing of initialization the initialization statements against the runtime data is performed.

After normal initialization processing completes, several reports are generated. The first report is the data set read report (see Example 5-17 on page 97). This report lists the initialization data sets that were read and the number of records that are processed from each data set.

If runtime data was obtained, the resource usage information is summarized (see Example 5-19 on page 99). This information is based on the system that is running at the time the initialization data set checker was run. The details of which system supplied the data also is provided in the summary report (see Example 5-20 on page 99).

A section of the report is reserved to provide recommendations for minimum settings for several resources. This value is based on reviewing the current usage ration of resources per job and projecting what is needed if the job limit is reached.

The summary report returns information about the JES2 instance that was verified. It includes the JES2 member name, node name, and XCF group name that is derived from the initialization data sets.

At the end of the report, the error summary provides a summary of any errors that are found during processing.

5.3 JES procedures

The JES2 procedure can point to a single PDS member, or several data sets. Members also can be concatenated to break up the JES2 initialization parameters into different members. For example, you might have one member that is common across the entire MAS, and a set of members that contains information that is specific to each system. Configurations with many NJE nodes often feature a member set that is set aside for NODE and CONNECT statements only.

The JES3 procedure points to a single member of a PDS, which can be overridden by replying M=xx to the IAT3012 message. INCLUDE statements can be used in the initialization deck to separate groups of statements into separate PDS members, if wanted.

5.3.1 JES2 procedure

It is common for JES2 to have a simpler procedure that is used to start it because the only JCL DD cards that are required are the HASPPARM and HASPLIST. A simpler JES2 procedure that is used to start a JES2 is shown in Example 5-21 on page 100.

Example 5-21 Sample basic JES2 initialization procedure

```
//JES2    PROC M=JES2IN00,M1=JES2IN&SYSCLONE,
//             PL=SYS1.JES2.PARMLIB,PROC=SYS1.PROCLIB
//IEFPROC EXEC PGM=HASJES20,TIME=1440,DPRTY=(15,15)
//HASPLIST  DD DDNAME=IEFRDER
//HASPPARM  DD DSN=&PL(&M),DISP=SHR
//          DD DSN=&PL(&M1),DISP=SHR
//PROC00    DD DSN=&PROC,DISP=SHR
```

JES2 can concatenate two or more members on the HASPPARM DD statement. Optionally, INCLUDE statements can be added to initialization deck data sets. As shown in Example 5-21, member JES2IN00 contains common statements and member JES2INxx contains z/OS image-specific statements. z/OS image-specific statements typically include devices, such as channel-attached printers, that can be physically attached to one z/OS image only.

System programmers must ensure all of the data sets that are referenced in the JES2 procedure are available or JES2 fails with a JCL error.

As shown in Example 5-22, the recommended method is to use dynamically define proclibs by way of PROCLIB statements in the initialization deck. The use of dynamic proclibs allows JES2 to start, even if a PROCLIB is missing or was mis-defined in the JES2 initialization statements. If a PROCLIB is not found during JES2 startup, a message can be issued and the operator then can correct or bypass the error.

The PROCLIB defined dynamic PROCLIB statements are shown in Example 5-22.

Example 5-22 Dynamic JES2 PROCLIBs

```
PROCLIB(PROC00) DD(1)=(DSN=SYS1.&SYSNAME..PROCLIB
                DD(2)=(DSN=SYS1.PROCLIB)
                DD(3)=(DSN=SYS1.IBM.PROCLIB)
PROCLIB(PROC01) DD(1)=(DSN=SYS1.LOGON.PROCLIB
                DD(2)=(DSN=SYS1.PROCLIB)
                DD(3)=(DSN=SYS1.IBM.PROCLIB)
PROCLIB(PROC02) DD(1)=(DSN=SYS1.STARTED.PROCLIB
PROCLIB(PROC04) DD(1)=(DSN=SYS1.FIN.PROCLIB
```

Dynamic proclibs can be added, modified, or removed by using the **`$ADD PROCLIB`**, **`$T PROCLIB`** or **`$DEL PROCLIB`** commands.

5.3.2 JES3 procedure

A JES3 procedure with all statements hardcoded is shown in Example 5-23.

Example 5-23 Typical JES3 procedure

```
//IEFPROC EXEC PGM=IATINTK,TIME=1440,PERFORM=255
//STEPLIB DD   DISP=SHR,DSN=SYS1.SIATLIB
//CHKPNT  DD   DISP=SHR,DSN=SYS1.JES3.CHECKPT
//CHKPNT2 DD   DISP=SHR,DSN=SYS1.JES3.CHECKPT2
//JES3IN  DD   DISP=SHR,DSN=SYS1.JES3.PARMLIB(JES3IN00)
```

The JES3IN DD statement on the JES3 procedure points to a single PDS member. The operator can select a different member by replying M=xx to the IAT3012 message.

JES3 supports INCLUDE statements so that the system programmer can break up the initialization deck into multiple members. For example, multiple members can be used to isolate statements that change frequently, such as DEVICE statements for printers or NJERMT statements, from the more critical parts of the deck.

JES3 supports the use of system symbols in its initialization statement, as does JES2. However, the Global system is the only system that reads the initialization statements in JES3, compared to JES2 where every system reads the initialization members. As a result, the use of system symbols in the initialization deck is less likely in a JES3 environment than in a JES2 environment.

The following DD statements also can be included in the JES3 procedure:

- JES3JCT
- JES3OUT
- JES3SNAP
- JESABEND
- IATPLBxx
- JES3DRDS

However, these statements typically are defined in DYNALLOC statements within the initialization deck rather than in the JES3 proc. The use of DYNALLOC allows the system to bypass missing data sets. If a data set that is referenced in the JES3 proc cannot be opened, JES3 fails with a JCL error.

A series of DYNALLOC statements for PROCLIBs in the JES3 initialization deck is shown in Example 5-24.

Example 5-24 Sample JES3 DYNALLOC statements to define PROCLIBs

```
* DYNALLOCS FOR PROCLIBS
* PROCLIBS ARE ACCESSED THROUGH THE CATALOG UNLESS UNIT AND VOLSER ARE CODED
* PROCLIB ST FOR STANDARD JOBS
DYNALLOC,DDN=IATPLBST,DSN=SYS1.SY1.PROCLIB
DYNALLOC,DDN=IATPLBST,DSN=SYS1.PROCLIB
DYNALLOC,DDN=IATPLBST,DSN=SYS1.IBM.PROCLIB
* PROCLIB 01 FOR TSO LOGONS
DYNALLOC,DDN=IATPLB01,DSN=SYS1.LOGON.PROCLIB
DYNALLOC,DDN=IATPLB01,DSN=SYS1.PROCLIB
DYNALLOC,DDN=IATPLB01,DSN=SYS1.IBM.PROCLIB
* PROCLIB 02 FOR STARTED TASKS
DYNALLOC,DDN=IATPLB02,DSN=SYS1.STARTED.PROCLIB
* PROCLIB FI FOR FINANCIAL JOBS
DYNALLOC,DDN=IATPLBFI,DSN=USER.FINANCE.PROCLIB,UNIT=3390,VOLSER=FIN001
```

The PROCLIB concatenations can be specified in the STANDARDS statement. INTPROC=ST specifies the standard PROCLIB concatenation for jobs that are entered by using the internal reader, STCPROC=02 specifies the concatenation for started task jobs, and TSOPROC=01 specifies the concatenation for TSO logons.

By using the definitions that are shown in Example 5-24, jobs that are submitted by the finance users can use their dedicated PROCLIB by specifying PROC=FI on the //*MAIN JECL statement in their jobs.

5.3.3 Other procedures

The following JES address spaces are started automatically at IPL time. No set up is needed. They shut down automatically when JES2 or JES3 ends:

- JESXCF: Common to both JESs
- JES2MON: JES2 Monitor address space
- JES2AUX: Auxiliary address space for JES2
- JES2EDS: JES2 Email Delivery Service address space
- JES3AUX: Auxiliary address space for JES3
- JES3DLOG: Hardcopy log for JES3

If a JESplex uses TCP/IP to drive NJE connections, one or more network server address spaces are started. A JES2 network server is named `*jesx*Snnn` where *jesx* is the name of the owning JES2 address space and `nnn` is the subscript on the NETSERV(nnn) statement.

For example, the first network server on a subsystem names JES2 is JES2S001. A JES3 network server can have any name, although a common name is JES3S001. A network server must be defined to the security product as a started task. IBM recommends the use of a common name pattern for network servers so that only one security profile is needed.

In addition, JES3 can have one or more CIFSS address spaces that are defined to offload some of the converter or interpreter workload. JES2 can process CI on any member of the JESplex and does not move this processing to a separate address space.

Both JESs can have one or more writer functional subsystems (FSSs) defined. These FSSs work the same way under JES3 and JES2.

An FSS can drive multiple printers. When a writer is called (JES3) or started (JES2), the system checks to see whether the appropriate FSS is running. If it is not, the system starts it automatically. An FSS cannot be started by using an operator command. The FSS ends automatically when the last printer it is driving is shut down.

5.4 Automation considerations

JES2 can be started without operator interaction by specifying PARM='WARM,NOREQ' on the JES2 **START** command. The NOREQ parameter relieves the operator from having to enter $S to start processing. This feature is the equivalent of coding PARM=NOREQ on the EXEC statement of the JES3 procedure to relieve the operator of having to enter *S JSS.

The JES3 start issues WTORs to determine the start type, and selects the initialization deck for a hot start with refresh, warm starts, or cold starts.

Any automation routines that manage JES3 startup, shutdown, failures, or restarts must be changed to address JES2 messages and commands.

Many automation routines key on JES3 initialization message IAT3100. These routines must be changed to JES2 message $HASP492.

5.5 JES2 policies

When started by the JES2 code, an installation-written exit routine for JES2 includes full access to various JES2 control blocks and JES2 internal functions. In this way, these JES2 exit interfaces provide ultimate flexibility in customizing JES2 processing.

To create an installation-written exit routine, JES2 control structures and JES2 internal processing knowledge and good programming skills are required. These exit routines must be checked whenever a change is introduced in a JES2 control block or internal function.

The JES2 policies provide an alternative way to customize JES2 processing without requiring in-depth programming knowledge, JES2 internal control structures, or an understanding of the JES2 internal processing logic.

You can formulate customization requirements in high-level terms based on general understanding of z/OS jobs requirements and their attributes. These requirements are defined to JES2 in a high-level human readable syntax. At a high level, a JES2 policy defines (in user-level terms) what JES2 must do in certain strategic points in JES2 processing.

At a conceptual level, each JES2 policy definition describes a condition that is a logical expression that determines whether this policy is applicable at a specific point in JES2 processing to a specific JES2 object, and one or more actions that must be performed if the condition is satisfied.

You can define multiple policies of the same type. Similarly, a single policy can feature multiple conditions that are defined, each with a set of associated actions. JES2 considers all policies and applies actions for all conditions that are satisfied. The choice between having many small policies with one or a few definitions of having fewer policies but with larger set of definitions is entirely based on convenience.

Externally, a JES2 policy definition is a JSON object that is in a human-readable editable z/OS data set.

A JES2 policy definition is imported into JES2 by a JES2 command and the policy becomes available for JES2 processing.

5.5.1 Policy definition syntax rules

JES2 policy definition is a JSON object. Each policy type has its own set of JSON names (entries) that can be used in a policy definition. However, JSON names are available that are common for all JES2 policies as well as syntax rules that apply to policies of all types.

JSON names must be coded exactly as specified. They are case-sensitive and do not allow leading or trailing blanks. However, character values that are entered for the JSON names in a policy definition are not case-sensitive and can contain any number of trailing or leading blanks for readability, except for character literal strings that are delimited by single quotation marks (apostrophes).

Attention: When coding a policy, it must be created in EBCDIC encoding by using the code page 1047 only. If not, an error as shown in the following example can occur during the import processing:

```
$POLICY IMPORT,PLCYLIB=POLICY00,MEMBER=JCONVERT
$HASP1600 POLICY IMPORT request accepted.
$HASP1630 JSON parser reported error 00000109, reason code 100. 030
           Unexpected token parsing JSON value at offset 338.
$HASP1631 Location of error:. 031
          "           "definitions"   : {"condition" : "Substr(JobNa"
          .............................*............................
$HASP1602 IMPORT request for policy *UNKNOWN failed.
```

Standard JSON names

Each JES2 policy must include the standard names defined that are listed in Table 5-1.

Table 5-1 Standard JSON names

JSON name	Description
“policyName” : “ *policy-name”*	Defines a 16-character policy name. Policy name must be unique. The *policy-name* must start with an alphabetic character and must contain only alphanumeric characters (standard JCL rules).
“policyType” : “ *policy-type* “	Defines a 16-character policy type and must contain one of the supported values. The *policy-type* determines a set of JES2 objects and their attributes that can be used in the policy and the phase of JES2 processing when this policy is considered. The *policy-type* also determines the supported syntax of the rest of the policy definition.

JSON name	Description
“policyVersion”: n	Defines a version of a policy. The policy version is a decimal number and must specify one of the supported versions. Supported versions vary by the policy type.

5.5.2 Policy type supported by JES2

The policy type that is listed in Table 5-2 is supported by JES2 to use policy definition.

Table 5-2 Supported JES2 Policy type

Policy type	Location
JobConversion	A policy with type JobConversion takes effect at the end of the job conversion phase and after all of the JCL for the job is processed.

JobConversion JSON names

To required standard JSON names, the policy type JobConversion accepts the following JSON names:

- “definitions” : *definitions array*

  The value of this JSON name is an array of JSON objects. Each object in this array defines one condition and one or more actions to be applied to the job if condition is satisfied. Each object in the array uses the following structure:

  - “condition” : “*expression*”

    The value of “condition” is a character string that defines a logical expression. The result of this expression must be a logical value, which is true or false. If result of condition expression evaluation is true, the condition is satisfied, and all actions that are defined for this condition are applied.

  - “actions” : *actions array*

    The value of this JSON name is an array of JSON objects. Each object in this array defines one action to be applied to the job. Each object in the array uses the following structure:

    - “action” : “*action-name*”

      The value of “action” is a name of one of the supported actions. The supported actions for policy type JobConversion are listed in Table 5-3.

Table 5-3 Actions that are supported by JobConversion policy type

Action	Description
CancelJob	Applying this action, the current job that is being processed is canceled. This action does not require any other JSON names.
HoldJob	Applying this action places the current job in the hold status. This action does not require any other JSON names.
Leave	Applying this action leaves the current policy entirely. Processing resumes on the next Policy in the concatenation.
LogMessage	Applying this action sends a message to syslog only. This action does not change any job attributes. This action requires the following JSON name to describe the message to be sent: “message” : “*expression”*

Action	Description
ModifyJob	Applying this action assigns new value to a specific job attribute. This action requires the following JSON names to describe the action: ▸ “attribute” : “*attribute-name*”[a] ▸ “value” : “*expression*”[b]
SendMessage	Applying this action sends a message to operator. This action does not change any job attributes. This action requires the following JSON name to describe the message to be sent: “message” : “*expression*”

a. The value of this JSON name is a character string with the name of the JES2 job attribute to be modified.

b. The value of this JSON name is a character string with an expression that defines new value for the attribute that is specified in the “attribute”. The result of this expression must be a value of the same data type as the JES2 job attribute to which it applies.

Attributes that are supported by ModifyJob action

When the action ModifyJob is used, some attributes (as indicated by the “attribute” JSON name) can be set to a new value that is obtained as result of the “value” JSON name. The job attributes that can be used or modified during the processing of the JobConversion policy are listed in Table 5-4.

Table 5-4 Attributes that are supported by JobConversion Policy

Attribute name	Data type	Modifiable	Description
JobAcct	Character list	No	List of accounting entries that are specified on the JOB JCL statement
JobClass	Character	Yes	Job class
JobHasAffinity (*member-list*)	Logical	No	Tests if job has affinity to one of the members that are provided in the member-list
JobIsHeld	Logical	No	Tests if job is held
JobName	Character	No	Job name
JobOwner	Character	No	User ID of job owner
JobPgmList	Character list	No	List of program names that are specified on PGM= keyword of EXEC JCL statements in the job[5.5.3a]
JobPrty	Numeric	Yes	Job priority
JobSubmitter	Character	No	User ID of job submitter
JobType	Character	No	Job type[b]
SysAff	Character list	Yes	Job affinity
SrvClass	Character	Yes	Service class of job
SchEnv	Character	Yes	Scheduling environment of job

a. With JOBDEF INTERPRET=INIT, the list does not include program names that are specified from the reference syntax, such as PGM=*.stepname.DDname. To obtain a full list of program names, use JOBDEF INTERPRET=JES.

b. 'JOB' - batch job, 'STC' - started task, 'JobGroup' – job group definition, 'TSU' - TSO user session.

5.5.3 Commands that are used to manage POLICYLIB

Several commands are available to manage POLICYLIB concatenations that are used by JES2 to select the policies to be processed.

For a policy to be processed by JES2, it must be in a PDS r UNIX PATH and then imported and enabled by JES2.

To run these functions, more commands are available that are specific to manage the policies.

Defining a new concatenation for JES2 policy import

The command **$ADD PLCYLIB** is used to define a new data set concatenation to be used as a source for importing external definition of JES2 policies.

The **$ADD POLICYLIB** command ensures only that the data sets specified can be allocated. It does not ensure that they exist or can be opened and are usable.

Example 5-25 shows sample output from a **$ADD PLCYLIB** command that was used to add the POLICY00 POLICYLIB concatenation to JES2.

Example 5-25 Sample of $ADD PLCYLIB command to add a POLICYLIB concatenation

```
$ADD PLCYLIB(POLICY00),DD(1)=(DSNAME=SYS1.JES2.PLCYLIB)
IEF196I IEF237I 9788 ALLOCATED TO $PL00013
$HASP737 PLCYLIB(POLICY00) 387
$HASP737 PLCYLIB(POLICY00)  DD(1)=(DSNAME=SYS1.JES2.PLCYLIB,
$HASP737                    VOLSER=BH5CAT)
```

Modifying a concatenation for JES2 policy import

The **$T PLCYLIB** command modifies the POLICYLIB concatenation that was created by a **$ADD** command. By using this command, new data sets can be added to a concatenation, or existing data sets replaced or deleted from the concatenation.

If a **$T PLCYLIB** command is entered with no operands, target concatenation is reallocated. This feature is useful to retry a previously failed allocation or to get updates to an existing data set.

Example 5-26 shows sample output of the **$T PLCYLIB** command that was used to add a UNIX PATH to the existent POLICY00 POLICYLIB concatenation.

Example 5-26 Sample of $T PLCYLIB command to add a PATH to an existent POLICYLIB

```
$T PLCYLIB(POLICY00),DD(2)=(PATH='/u/jes2/plcylib/policy00')
IEF196I IEF237I 9788 ALLOCATED TO $PL00014
IEF196I IGD103I SMS UNIX FILE ALLOCATED TO DDNAME SYS00010
IEF196I IEF285I   SYS1.JES2.PLCYLIB                             KEPT
IEF196I IEF285I   VOL SER NOS= BH5CAT.
$HASP737 PLCYLIB(POLICY00) 401
$HASP737 PLCYLIB(POLICY00)  DD(1)=(DSNAME=SYS1.JES2.PLCYLIB,
$HASP737                    VOLSER=BH5CAT),
$HASP737                    DD(2)=(PATH=/u/jes2/plcylib/policy00
$HASP737                    )
```

Displaying a concatenation for JES2 policy import

The command **$D PLCYLIB** is used to display one or more existing POLICYLIB concatenations that are used by JES2 to import policies.

Example 5-27 shows sample output from a **$D PLCYLIB** command that was used to display information about the existing POLICY00 POLICYLIB concatenation.

Example 5-27 Sample of $D PLCYLIB command to display an existent POLICYLIB

```
$D PLCYLIB
$HASP737 PLCYLIB(POLICY00) 407
$HASP737 PLCYLIB(POLICY00)  DD(1)=(DSNAME=SYS1.JES2.PLCYLIB,
$HASP737                    VOLSER=BH5CAT),
$HASP737                    DD(2)=(PATH=/u/jes2/plcylib/policy00
$HASP737                    )
```

5.5.4 Commands that are used to manage policies

In this section, we describe commands that can be used to manage policies.

Importing a new JES2 policy

The command **$POLICY IMPORT** is used to import a policy from an existing POLICYLIB concatenation to JES2 processing. This command processes a JES2 policy definition in human-readable external representation (in JSON notation), converts the policy into internal representation that is used by JES2, and makes the policy available for JES2 processing.

At the import process by JES2, the *policy-name* that is used on policyName JSON name is set to imported policy. On further commands to manage the policy, the policyName that is associated with the policy must be used.

Example 5-28 shows a sample output of the **$POLICY IMPORT** command that was used to import and enable the JES2 Policy type JobConversion named JCONVERT. This policy was coded in a UNIX path that was concatenated to PLCYLIB POLICY00.

Example 5-28 Sample of $POLICY IMPORT command used to import the JES2 Policy JCONVERT

```
$POLICY IMPORT,PLCYLIB=POLICY00,MEMBER=JCONVERT
$HASP1600 POLICY IMPORT request accepted.
$HASP1603 Validation of policy JCONVERT type JobConversion is 115
          complete.
$HASP1611 Policy JCONVERT type JobConversion saved in the JES2 116
          checkpoint.
$HASP1614 Policy JCONVERT type JobConversion added to runtime 117
          repository.
$HASP1601 IMPORT policy JCONVERT request complete.
```

Enabling a JES2 policy

The command **$POLICY ENABLE** is used to change JES2 policy to an enabled state.

If the policy import command that is used to import the policy was not coded to include the ENABLE subparameter, the policy is imported in enabled state and is immediately available for JES2 processing. This option is the default option to ENABLE subparameter.

Example 5-29 shows a sample output of the **$POLICY ENABLE** command that can be used to enable a policy import to JES2 in disable state. The **$POLICY ENABLE** can be issued to a policy that is enabled without error.

Example 5-29 Sample of $POLICY ENABLE command used to enable the JES2 Policy JCONVERT

```
$POLICY ENABLE,NAME=JCONVERT
$HASP1601 ENABLE policy JCONVERT request complete.
```

Disabling a JES2 policy

The command **$POLICY DISABLE** is used to change JES2 policy to a disabled state.

This command does not remove policy from JES2 configuration, but JES2 does not use this policy until it is enabled again by the **$POLICY ENABLE** command

Example 5-30 shows sample output of the **$POLICY DISABLE** command that is used to disable the JES2 Policy JCONVERT.

Example 5-30 Sample of $POLICY DISABLE command issued to disable a JES2 Policy

```
$POLICY DISABLE,NAME=JCONVERT
$HASP1623 Policy JCONVERT type JobConversion disabled.
$HASP1601 DISABLE policy JCONVERT request complete.
```

Deleting JES2 policy

The command **$POLICY DELETE** is used to delete JES2 policy from JES2 configuration. When the policy is deleted, it is no longer available for JES2 processing. To reuse it, a new **$POLICY IMPORT** command must be issued.

Example 5-31 shows sample output of the **$POLICY DELETE** command that is used to remove the JES2 Policy JCONVERT from JES2.

Example 5-31 Sample of $POLICY DELETE command used to remove a JES2 policy

```
$POLICY DELETE,NAME=JCONVERT
$HASP1616 Policy JCONVERT type JobConversion deleted.
$HASP1601 DELETE policy JCONVERT request complete.
```

5.5.5 JES2 policy example

Example 5-32 shows sample output of JES2 policy JCONVERT type JobConversion that is used to run the related tests with the JES2 Policy functionality.

The policy was coded by using EBCDIC 1047 code page in a UNIX path.

Example 5-32 Sample of JES2 Policy type JobConversion

```
{
 "policyName"    : "JCONVERT",
 "policyVersion" : 1,
 "policyType"    : "JobConversion",
 "definitions"   :
 [
  {
   "condition" : "Substr(JobName,1,4) = '#RED'",
   "actions"   :
    [
     {
      "action"    : "ModifyJob",
      "attribute" : "SYSAFF",
      "value"     : "List('SC74')"
     },
     {
      "action"    : "ModifyJob",
      "attribute" : "JOBPRTY",
      "value"     : "15"
     },
     {
      "action"    : "SendMessage",
      "message"   : "'PLCY001I '||JobType||' '||JobName
       ||' now have priority '||String(JobPrty)
       ||' and Affinity to '||String(SysAff)"
     }
    ]
  }
 ]
}
```

Example 5-33 shows the sample JCL that was used to validate the processing of JES2 Policy JCONVERT created.

The job selection is based on jobname staring with #RED.

Example 5-33 Sample JCL used to test the JES2 Policy JCONVERT

```
//#REDJES2 JOB (),'ITSO REDBOOKS',CLASS=A,MSGCLASS=X,
//         MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID
//*
//STEP01   EXEC PGM=IEFBR14
```

Example 5-34 shows the messages that are issued to Operator’s Console during the processing of a job that is selected by the JES2 policy and complementary exits 11 and 12 that are installed to manage the spool partitioning processing. For more information about exits 11 and 12, see Appendix E, “SPOOL partitioning exits sample code” on page 233.

Example 5-34 Messages issued to OPERLOG during Job processing and by JES2 Policy JCONVERT

```
$EXT1108I JOBCLASS A ALLOWED TO USE SPOOL PARTITION BY JOB #REDJES2
ON SYSID SC74
IEF196I $EXT1108I JOBCLASS A ALLOWED TO USE SPOOL PARTITION BY JOB
IEF196I #REDJES2 ON SYSID SC74
$EXT1101I JOB #REDJES2 SELECTED TO USE SPOOL PARTITION ON SYSID SC74
IEF196I $EXT1101I JOB #REDJES2 SELECTED TO USE SPOOL PARTITION ON
SYSID
IEF196I SC74
$EXT1103I VOLUME BH5SP1 ADDED TO JOB #REDJES2 AS OVRFLOW SPOOL VOLUME
IEF196I $EXT1103I VOLUME BH5SP1 ADDED TO JOB #REDJES2 AS OVRFLOW SPOOL
IEF196I VOLUME
$HASP100 #REDJES2 ON INTRDR      ITSO REDBOOKS         FROM TSU01402
LPRES3
IRR010I  USERID LPRES3   IS ASSIGNED TO THIS JOB.
PLCY001I Job #REDJES2 now have priority 15 and Affinity to (SC74)
```

Part 2

Use case study

In this part, we describe a sample case study that is based on a customer experience that was completed in 2015.

6

Customer experience case study

This chapter describes the experiences of a mid-sized business customer when they chose to migrate from JES3 to JES2.

This chapter includes the following topics:

- 6.1, “Migration steps overview” on page 114
- 6.2, “Planning and assumptions” on page 115
- 6.3, “JES2 system design” on page 124
- 6.4, “Educating stakeholders” on page 135
- 6.5, “Removing and replacing JES3 exits” on page 137
- 6.6, “Transforming JES3 special functions” on page 143
- 6.7, “Transforming JCL and JECL” on page 150
- 6.8, “Migrating system automation” on page 153
- 6.9, “Migrating security” on page 157
- 6.10, “Migrating your printer” on page 167
- 6.11, “Performance experience” on page 170
- 6.12, “Hints and tips” on page 172
- 6.13, “Ready to migrate” on page 180

6.1 Migration steps overview

An overview of the necessary tasks that must be done during the JES3-to-JES2 migration project are listed in Table 6-1. Some of the listed tasks might not be included in your migration project based on your system environment.

Table 6-1 Project tasks overview

Project task	Target	Comments
Project management	Covers all types of project management, such as: ► Managing project ► Stakeholder plannings ► Communication plan ► Resource plan ► Vacation plan ► Education plan ► Organize meetings	N/A
Detachment JES3 Exits	► Identify all used JES3 exits ► Identify all used PSF printing exits ► Create migration strategy (removal or transformation to exits) ► Create needed JES2 exits	We recommend starting the removal of JES3 exits as soon as possible.
Detachment JES3 special functions	► Replace JES3 DEADLINE ► Replace JES3 DJC ► Disable MDS ► Establish JES2 equivalent for all functions ► Disable JES3 DLOG and activate OPERLOG	We recommend starting the process to disable JES3 functions as soon as possible before the project starts. For DLOG deactivation, you must rewrite some programs that are using DLOG.
Changes for JCL/JECL	► Convert production JCL ► Provide tool to convert user JCL	Try to align your production JCL to work with both JES versions. This alignment can be done as soon as possible and features no dependencies.
Security changes	► Add JES2 security profiles to RACF ► Assign permissions to stakeholders ► Add SDSF/EJES profiles to RACF ► Add profile definitions for changed printer names	The RACF permissions that are defined for JES2 should match to profiles that being made or exist for SDSF/EJES.
System automation	► Analyze JES3 automation that is in place ► Setup new JES2 automation	This task can take some time because you must review all automation points that you defined for JES3. Then, you must decide whether this task must be transferred to a JES2 solution.

Project task	Target	Comments
JES printing	► Analyze printing environment ► Code new JES2 print exits ► Adjust procedures for printing that uses JES3 commands	Consider some extra time to create JES2 exits and conduct printing tests. We faced some unplanned issues that needed to be sorted.
JES2 setup	► Define JES2 setup according to your company defaults ► Calculate needed storage and request DASD for all systems ► Request all other resources that you need (HLQ, RACF, and STC) ► Request TCP/IP and firewall settings for NJE ► Start JES2 on all environments to verify the function	This task can be done during normal system operation time. JES2 can be run in parallel to JES3 as secondary subsystem. You should calculate some time to define all the standards you need for the JES2.
General things to do	► Provide education material ► Conduct education sessions ► Quit your JES3 license	These tasks are an important part of the project. To get the most possible acceptance for the project from all stakeholders, you should conduct information session at the earliest point in the start process. You also should frequently update stakeholders.

Most of the tasks that are listed in Table 6-1 on page 114 can run in parallel and have almost no dependencies to each other.

6.2 Planning and assumptions

The case study that is described in this section is based on experience from a real customer situation with their JES3 migration. The case study was completed with z/OS V2R1and was now updated to the z/OS V2R4 level. This configuration makes a JES3 Migration much easier for customers. The customer runs eight sysplexes that vary 2 - 8 systems in the sysplex.

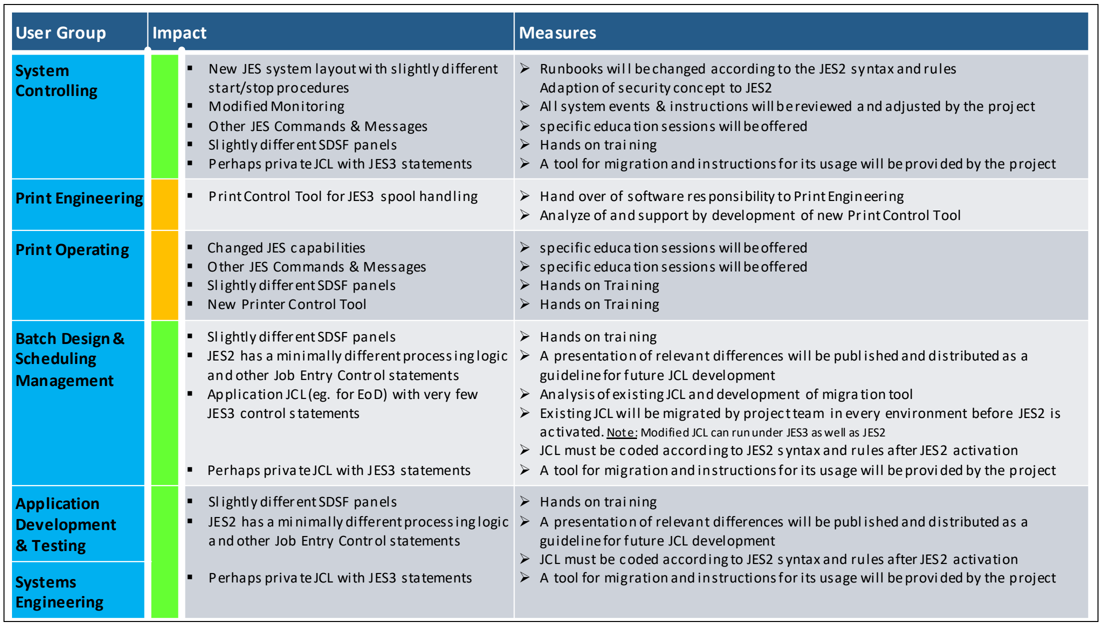
To enable all functions of z/OS V2R4, complete the following steps:

1. Migrate your JES2 Checkpoint to the z22 mode by using the **`$ACTIVATE`** command.
2. Enable SPOOL compression and encryption by using the **`$TSPOOLDEF,ADVF=ENABLED`** command.
3. Enable JES2 disk reader support by adding one or more SUBMITLIB and SUBMITRDR statements to your JES2 initialization member.
4. Use the new supported JES3 JECL ROUTE XEQ by enabling that support in your JES2 initialization member by using the following statements:
   - INPUTDEF JES3JECL=SUPPORT
   - JECLDEF JES3=(ROUTE=PROCESS)

The sysplex environment contains the classic structure from sandbox sysplex over test sysplexes to the production sysplex. The entire migration was done in approximately six months.

Identifying stakeholders

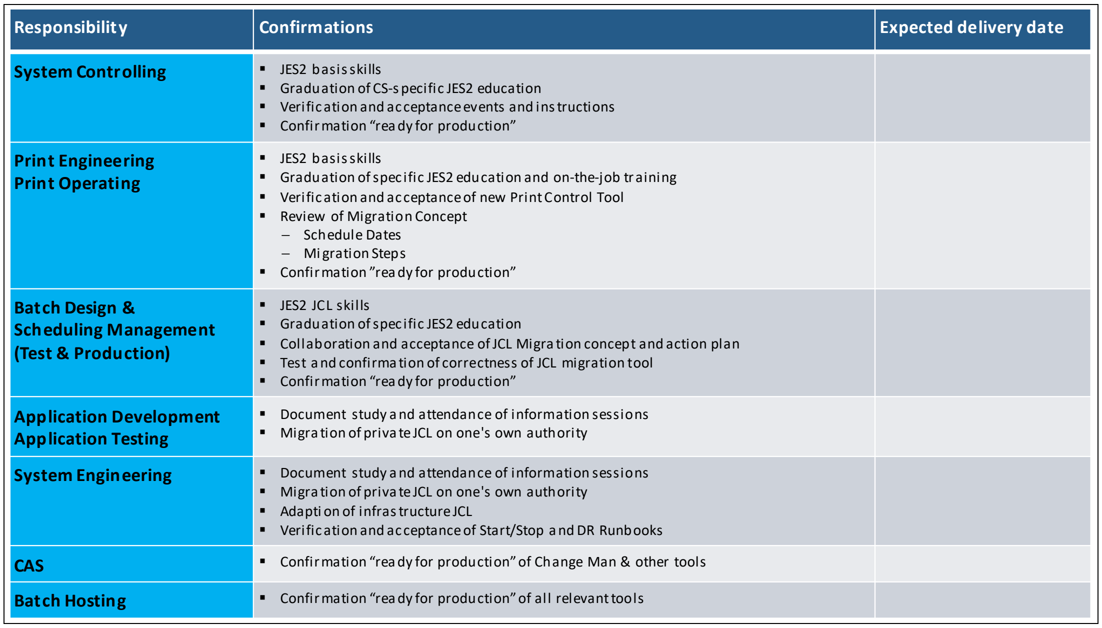
One of the first tasks that must be done is to identify all affected stakeholders. An example of an overview of the stakeholders, the possible affect, and the open tasks for that group of stakeholders is shown in Figure 6-1.



User Group	Impact		Measures
System Controlling	🟩	▪ New JES system layout with slightly different start/stop procedures ▪ Modified Monitoring ▪ Other JES Commands & Messages ▪ Slightly different SDSF panels ▪ Perhaps private JCL with JES3 statements	➢ Runbooks will be changed according to the JES2 syntax and rules Adaption of security concept to JES2 ➢ All system events & instructions will be reviewed and adjusted by the project ➢ specific education sessions will be offered ➢ Hands on training ➢ A tool for migration and instructions for its usage will be provided by the project
Print Engineering	🟧	▪ Print Control Tool for JES3 spool handling	➢ Hand over of software responsibility to Print Engineering ➢ Analyze of and support by development of new Print Control Tool
Print Operating	🟧	▪ Changed JES capabilities ▪ Other JES Commands & Messages ▪ Slightly different SDSF panels ▪ New Printer Control Tool	➢ specific education sessions will be offered ➢ specific education sessions will be offered ➢ Hands on Training ➢ Hands on Training
Batch Design & Scheduling Management	🟩	▪ Slightly different SDSF panels ▪ JES2 has a minimally different processing logic and other Job Entry Control statements ▪ Application JCL (eg. for EoD) with very few JES3 control statements ▪ Perhaps private JCL with JES3 statements	➢ Hands on training ➢ A presentation of relevant differences will be published and distributed as a guideline for future JCL development ➢ Analysis of existing JCL and development of migration tool ➢ Existing JCL will be migrated by project team in every environment before JES2 is activated. Note: Modified JCL can run under JES3 as well as JES2 ➢ JCL must be coded according to JES2 syntax and rules after JES2 activation ➢ A tool for migration and instructions for its usage will be provided by the project
Application Development & Testing	🟩	▪ Slightly different SDSF panels ▪ JES2 has a minimally different processing logic and other Job Entry Control statements ▪ Perhaps private JCL with JES3 statements	➢ Hands on training ➢ A presentation of relevant differences will be published and distributed as a guideline for future JCL development ➢ JCL must be coded according to JES2 syntax and rules after JES2 activation ➢ A tool for migration and instructions for its usage will be provided by the project
Systems Engineering			

Figure 6-1 Stakeholder sample planning

After the stakeholders are identified, you assign all the necessary tasks to them and publish a final date of completion to bring the project to success. An example of tasks that must be done by every stakeholder is shown in Figure 6-2.

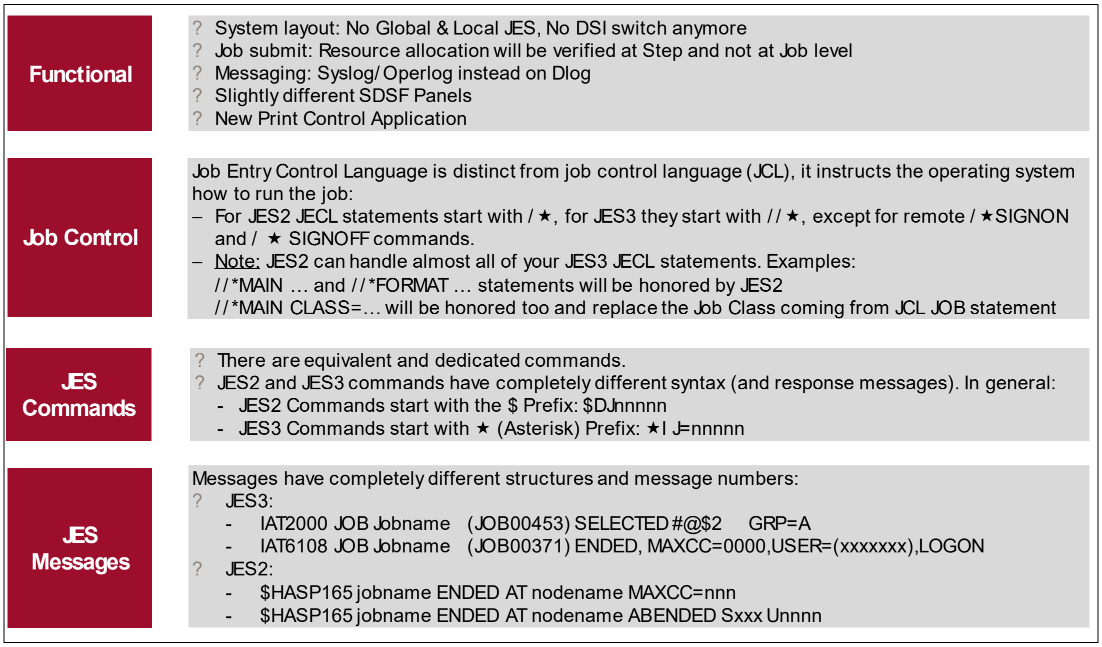


Responsibility	Confirmations	Expected delivery date
System Controlling	▪ JES2 basis skills ▪ Graduation of CS-specific JES2 education ▪ Verification and acceptance events and instructions ▪ Confirmation “ready for production”	
Print Engineering Print Operating	▪ JES2 basis skills ▪ Graduation of specific JES2 education and on-the-job training ▪ Verification and acceptance of new Print Control Tool ▪ Review of Migration Concept – Schedule Dates – Migration Steps ▪ Confirmation ”ready for production”	
Batch Design & Scheduling Management (Test & Production)	▪ JES2 JCL skills ▪ Graduation of specific JES2 education ▪ Collaboration and acceptance of JCL Migration concept and action plan ▪ Test and confirmation of correctness of JCL migration tool ▪ Confirmation “ready for production”	
Application Development Application Testing	▪ Document study and attendance of information sessions ▪ Migration of private JCL on one's own authority	
System Engineering	▪ Document study and attendance of information sessions ▪ Migration of private JCL on one's own authority ▪ Adaption of infrastructure JCL ▪ Verification and acceptance of Start/Stop and DR Runbooks	
CAS	▪ Confirmation “ready for production” of Change Man & other tools	
Batch Hosting	▪ Confirmation “ready for production” of all relevant tools	

Figure 6-2 Expectations from stakeholders

Main differences overview

To give all stakeholders a brief overview what is going to be changed during migration, it might be helpful to provide a one-pager to all stakeholders. Use the chart that is shown in Figure 6-3 to provide a brief overview to your stakeholders about that main differences between both JES versions.



Functional

? System layout: No Global & Local JES, No DSI switch anymore
? Job submit: Resource allocation will be verified at Step and not at Job level
? Messaging: Syslog/ Operlog instead on Dlog
? Slightly different SDSF Panels
? New Print Control Application

Job Control

Job Entry Control Language is distinct from job control language (JCL), it instructs the operating system how to run the job:
– For JES2 JECL statements start with /★, for JES3 they start with //★, except for remote /★SIGNON and / ★ SIGNOFF commands.
– Note: JES2 can handle almost all of your JES3 JECL statements. Examples:
//*MAIN … and //*FORMAT … statements will be honored by JES2
//*MAIN CLASS=… will be honored too and replace the Job Class coming from JCL JOB statement

JES Commands

? There are equivalent and dedicated commands.
? JES2 and JES3 commands have completely different syntax (and response messages). In general:
- JES2 Commands start with the $ Prefix: $DJnnnnn
- JES3 Commands start with ★ (Asterisk) Prefix: ★I J=nnnnn

JES Messages

Messages have completely different structures and message numbers:
? JES3:
- IAT2000 JOB Jobname (JOB00453) SELECTED #@$2 GRP=A
- IAT6108 JOB Jobname (JOB00371) ENDED, MAXCC=0000,USER=(xxxxxxx),LOGON
? JES2:
- $HASP165 jobname ENDED AT nodename MAXCC=nnn
- $HASP165 jobname ENDED AT nodename ABENDED Sxxx Unnnn

Figure 6-3 Overview JES comparison

Third-party products

Almost every customer is using third-party software on their mainframes. All of this software must be checked for the JES2 compatibility and adjusted so that it is compatible, if necessary. For this task, contact the software vendor as soon as possible for written proof of compatibility.

CONTROL-M

If you use CONTROL-M instead of Tivoli Workload Scheduler for controlling your BATCH processing, you must tell CONTROL-M that it must operate with JES2. A portion of the IOAPARM data set of CONTROL-M with the JES3 definitions is shown in Example 6-1.

Example 6-1 CONTROL-M JES3 support

```
*-------------------------------------
*   JES parameters
*-------------------------------------
JES      JESTYPE=JES3,            JES type installed
         JESCHAR=*,               JES command character
         JESREL=,                 JES release
         d=                 Method of issuing JES3 commands
```

To activate JES2 support for CONTROL-M, replace JESTYPE and JESCHAR statements to support JES2. The portion of the IOAPARM member with JES2 support is shown in Example 6-2.

Example 6-2 CONTROL-M JES2 support

```
*-------------------------------------
*   JES parameters
*-------------------------------------
JES      JESTYPE=JES2,           JES type installed
         JESCHAR=$,              JES command character
         JESREL=,                JES release
         JES3CMD=                Method of issuing JES3 commands
```

EJES

The most common third-party product for JES3 users is EJES. To enable JES2 support in EJES, you can APPLY the EJES$ENV USERMOD that is included with the product. The following USERMODs are suitable for JES2 and JES3 products:

EJES$ENV USERMOD for the EJES environment for JES2 and JES3

EJES$EN2 USERMOD for the EJES environment for JES2 only

EJES$EN3 USERMOD for the EJES environment for JES3 only

This job installs the USERMOD that generates the system environment tables. EJES$ENV is used when support for JES2 and JES3 is being generated. The EJES$ENV job can be useful during migration to support both JES versions. After the final deactivation of JES3, you can update your SMP/E environment to use EJES$EN2 only.

In sysplex installations, EJES is using a so-called Coordination Address Space (CAS) server to exchange data between all systems within a sysplex. This CAS needs a global data set (see Example 6-3), which is shared by all EJES participants under JES3.

Example 6-3 JES3 EJES CAS

```
//EJESCAS  PROC GDSN='JES3#A.RZO.PO.EJES.GLOBAL.DATA',
//             CDSN='JES3#A.RZO.PO.INISH',
//             MBR=EJESCASC,
//             PRTY='(o)',
//             SSYS=EJES,
//             SC=T
//EJESCAS  EXEC PGM=EJESCAS,TIME=1440,DPRTY=&PRTY,
//            PARM='CASKEY=&SSYS'
//GBLDATA  DD DSN=&GDSN,DISP=SHR
//CASCONFG DD DSN=&CDSN.(&MBR),DISP=SHR
//SYSABEND DD SYSOUT=&SC
//SYSOUT   DD SYSOUT=*
//
```

When EJES is used with JES2, the CAS shared data set is no longer needed and can be removed from EJES CAS start procedures in your installation. An example of an EJES CAS started task for JES2 working with default values is shown in Example 6-4.

Example 6-4 JES2 EJES CAS

```
//EJESCAS  PROC
//EJESCAS  EXEC PGM=EJESCAS,TIME=1440
//SYSABEND DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//
```

Attention: You must plan and adjust your EJES/SDSF security definitions because of added, changed, or deleted panels or fields that can be typed over in panels. These modified settings must be in line with your JES2 security settings.

ACF2

The customer installation was protected by ACF2 instead of RACF during migration to JES2. To establish the JES2 support in ACF2, you must activate several exit routines according to the ACF2 installation guide. The requested ACF2 user exits that must be loaded in JES2 are shown in Example 6-5.

Example 6-5 ACF2 JES2 exits

```
LOAD(ACFJ2ITF) STOR=CSA                       /* ACF2/JES2 interface */
EXIT2   ROUTINE=ACFEXIT2                    /* job card scan routine */
EXIT4   ROUTINE=ACFEXIT4                    /* jcl card scan routine */
EXIT20  ROUTINE=ACFEXT20                       /* end-of-rdr manager */
EXIT24  ROUTINE=ACFEXT24                 /* Post-Initialization Exit */
EXIT26  ROUTINE=ACFEXT26                         /* Termination Exit */
EXIT31  ROUTINE=ACFEXT31             /* SSI Data Set Allocation Exit */
EXIT34  ROUTINE=ACFEXT34           /* SSI Data Set Unallocation Exit */
EXIT46  ROUTINE=ACFEXT46                        /* NJE Transmit Exit */
EXIT50  ROUTINE=ACFEXT50    /* end-of-rdr manager - user environment */
EXIT52  ROUTINE=ACFEXT52 /* job card scan routine - user environment */
EXIT54  ROUTINE=ACFEXT54 /* jcl card scan routine - user environment */
EXIT56  ROUTINE=ACFEXT56     /* NJE Transmit Exit - user environment */
EXIT225 ROUTINE=ACFEX225                 /* subtask attach/post rtne */
EXIT227 ROUTINE=ACFEX227,DISABLE            /* debug message routine */
```

ThruPut Manager

In the customer environment, a case study for using Compuware ThruPut Manager was conducted at the time of migration. ThruPut Manager can help automatically and intelligently optimize your batch processing by balancing workload and improving batch throughput.

By using ThruPut Manager, customers can perform the following tasks:

- Better prioritize batch processing based on business policies and goals
- Automatically select the most urgent jobs first without system overload
- Verify that jobs include the resources that they need
- Proactively manage resource contention between jobs

The JES2 initialization statements of ThruPut manager are shown in Figure 6-4 on page 120. Most of the changes are the loading and activation of several JES2 exits that are used by ThruPut manager. In the customer case, we combined some ACF2 exits with another ThruPut manager exit (see Figure 6-4 for JES2 exit 2, 4, 24, 52, and 54).

```
/*-----------------------------------------------------------------*/
/* Load JES2 Exits requested by ThruPut Manager                    */
/* and set default initialization parameter using TMPARM           */
/*-----------------------------------------------------------------*/
LOADMOD(DTMJ2MV7) STORAGE=PVT   /* LOAD MAIN TASK EXIT MODULE      */
LOADMOD(DTMJ2SV7) STORAGE=CSA   /* LOAD SSSM EXIT MODULE           */
EXIT(2)  ROUTINE=(DTMJ2X02,ACFEXIT2),STATUS=ENABLED,TRACE=NO
EXIT(4)  ROUTINE=(DTMJ2X04,ACFEXIT4),STATUS=ENABLED,TRACE=NO
EXIT(5)  ROUTINE=(DTMJ2X05),STATUS=ENABLED,TRACE=NO
EXIT(7)  ROUTINE=(DTMJ2X07),STATUS=ENABLED,TRACE=NO
EXIT(8)  ROUTINE=(DTMJ2X08),STATUS=ENABLED,TRACE=NO
EXIT(10) ROUTINE=(DTMJ2X10),STATUS=ENABLED,TRACE=NO
EXIT(14) ROUTINE=(DTMJ2X14),STATUS=ENABLED,TRACE=NO
EXIT(19) ROUTINE=(DTMJ2X19),STATUS=ENABLED,TRACE=NO
EXIT(24) ROUTINE=(DTMJ2X24,ACFEXT24),STATUS=ENABLED,TRACE=NO
EXIT(49) ROUTINE=(DTMJ2X49),STATUS=ENABLED,TRACE=NO
EXIT(51) ROUTINE=(DTMJ2X51),STATUS=ENABLED,TRACE=NO
EXIT(52) ROUTINE=(DTMJ2X52,ACFEXT52),STATUS=ENABLED,TRACE=NO
EXIT(54) ROUTINE=(DTMJ2X54,ACFEXT54),STATUS=ENABLED,TRACE=NO
TMPARM COMCHAR=/,          /* TM COMMUNICATION CHARACTER           */
       TYPHOLD,            /* TYPRUN=HOLD JOBS 1ST ANALYZED THEN HELD*/
       TYPSCAN,            /* TYPRUN=SCAN JOBS 1ST ANALYZED THEN HELD*/
       CUSTOM=(UM0035),    /* RETAIN PRE-TM EXECUTION CLASS IN SMF30 */
       OPTIONS=(OFF(DCS)), /* DISABLE DATASET CONTENTION SERVICES    */
       TMINITS=(3,12,,2)   /* 3X100 $$TM DYNAMIC INITS MAX PER LPAR  */
                           /* 12 HRS OR 1000 JOBS BEFORE INIT RECYCLE*/
                           /* 2X DYNAMIC ANALYZERS INITS PER 100 JOBS*/
```

Figure 6-4 Sample JES2 TM initialization

> **Attention:** All JES2 exits must be compiled and linked with the JES2 version that is running.

All JES2 exits have one thing in common: The exits must be compiled or linked with the same JES2 version they plan to use. A mismatched JES2 stops initialization (see Figure 6-5).

```
 $HASP466 INCLUDE    STMT   217  LOADMOD(DTMJ2MV7) STORAGE=PVT
 $HASP466 INCLUDE    STMT   217    /* LOAD MAIN TASK EXIT
 $HASP466 INCLUDE    STMT   217  MODULE        */
 $HASP003 RC=(34),LOADMOD(DTMJ2MV7)  - DTMJ2MV7 - MODULE VERSION
 $HASP003        IS INVALID, EXPECTED-z/OS 2.3 ACTUAL-z/OS 2.2
*3587 $HASP469 REPLY PARAMETER STATEMENT, CANCEL, OR END
```

Figure 6-5 Example of wrong JES2 exit version

You now must re-create the all of the JES2-depended exits with the correct version and reload the exit under JES2. Also, strong dependency exists between TM and JES2 control blocks. Therefore, after any JES2 maintenance that affects JES2 control blocks, you must recompile the following Thruput manager modules (z/OS version and maintenance level depended exist):

- DTMJ2DB7
- DTMJ2GO7
- DTMJ2MV7
- DTMJ2SV7

Consider moving these z/OS level-dependent modules to a separate library and concatenate them in front of the Thruput manager base load library DTMLINK.

Thruput Manager must work correctly to separate job classes. Some examples from a customer's installation are shown in Figure 6-6. The Jobclass names can vary in other installations.

```
JOBCLASS(TMA,TMG,TMO) COMMAND=IGNORE,
         MSGCLASS=E,         /* DEFAULT MESSAGE CLASS                */
         MODE=JES            /* JES INIT'S                           */
```

Figure 6-6 Jobclasses for Thruput manager

After restarting JES2 with the new exits and jobclass definitions, you must complete the initial configuration. The Thruput manager initial commands that are used to configure the basic jobclass selection are shown in Example 6-6.

Example 6-6 Basic TM commands for jobclass setup

```
$MJ,TM,CLASS,SET,Analyze(TMA)
$MJ,TM,CLASS,SET,General_Services(TMG)
$MJ,TM,CLASS,SET,On_Demand(TMO)
$MJ,TM,CLASS,DELETE(ALL)
```

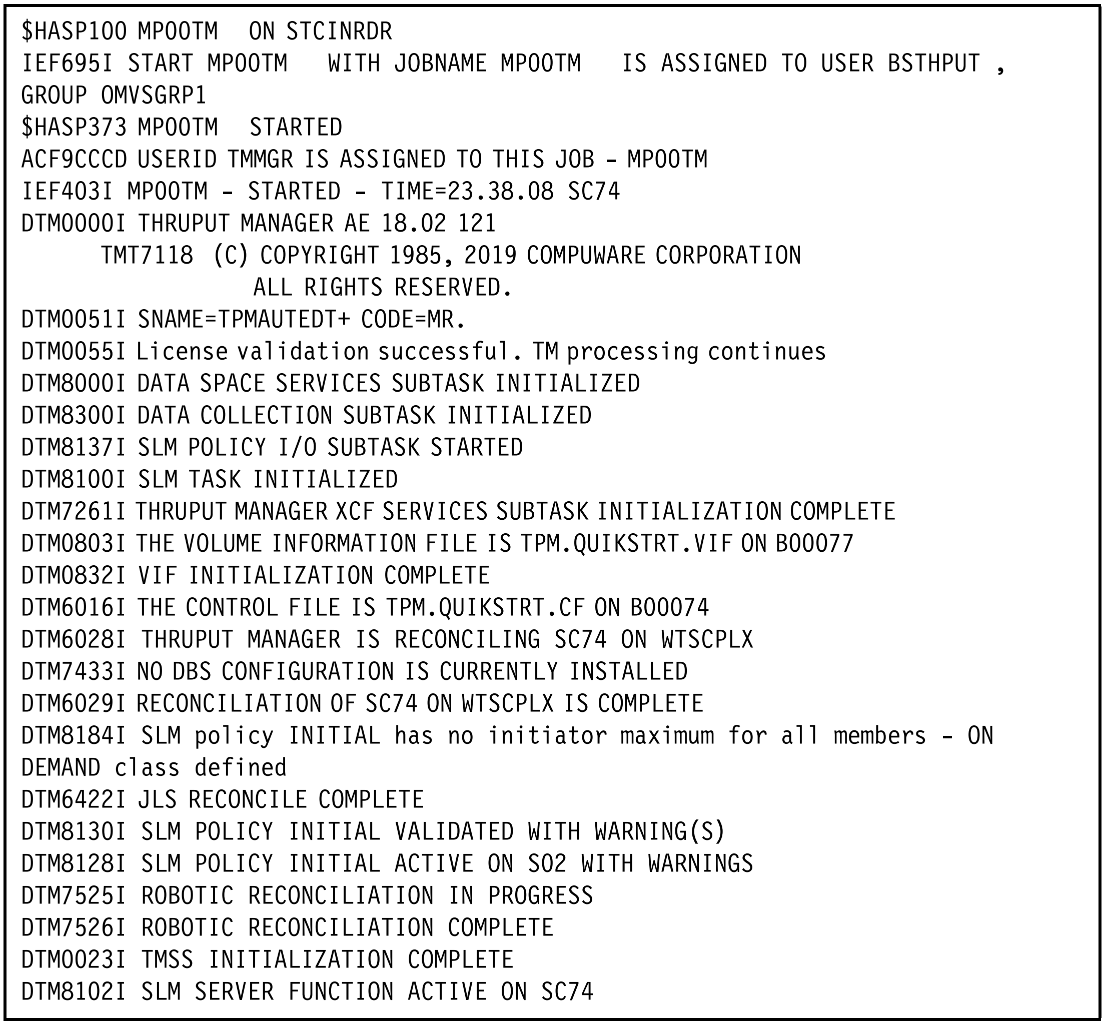
The response of Thruput manager after defining all jobclasses is shown in Figure 6-7. The jobclasses can vary in your environment.

```
$MJ,TM,CLASS,D
DTM3233I TM CLASS LIST 039
Analysis............ TMA
Deferred............ None
Selectable.......... TMA,TMG,TMO
Exempt.............. A,M,M1,P0,P1,P2,P3,S0,S1,S2
On_Demand........... TMO
Production_Services. None
General_Services.... TMG
Default............. None
$MJ,START
```

Figure 6-7 TM setup verification

After these steps are completed, you now can start Thruput Manager in your system.

An example of starting the Thruput manager is shown in Figure 6-8. Upon completion, you can use the product in your installation.



```
$HASP100 MP00TM   ON STCINRDR
IEF695I START MP00TM   WITH JOBNAME MP00TM   IS ASSIGNED TO USER BSTHPUT ,
GROUP OMVSGRP1
$HASP373 MP00TM   STARTED
ACF9CCCD USERID TMMGR IS ASSIGNED TO THIS JOB - MP00TM
IEF403I MP00TM - STARTED - TIME=23.38.08 SC74
DTM0000I THRUPUT MANAGER AE 18.02 121
      TMT7118  (C) COPYRIGHT 1985, 2019 COMPUWARE CORPORATION
                   ALL RIGHTS RESERVED.
DTM0051I SNAME=TPMAUTEDT+ CODE=MR.
DTM0055I License validation successful. TM processing continues
DTM8000I DATA SPACE SERVICES SUBTASK INITIALIZED
DTM8300I DATA COLLECTION SUBTASK INITIALIZED
DTM8137I SLM POLICY I/O SUBTASK STARTED
DTM8100I SLM TASK INITIALIZED
DTM7261I THRUPUT MANAGER XCF SERVICES SUBTASK INITIALIZATION COMPLETE
DTM0803I THE VOLUME INFORMATION FILE IS TPM.QUIKSTRT.VIF ON B00077
DTM0832I VIF INITIALIZATION COMPLETE
DTM6016I THE CONTROL FILE IS TPM.QUIKSTRT.CF ON B00074
DTM6028I THRUPUT MANAGER IS RECONCILING SC74 ON WTSCPLX
DTM7433I NO DBS CONFIGURATION IS CURRENTLY INSTALLED
DTM6029I RECONCILIATION OF SC74 ON WTSCPLX IS COMPLETE
DTM8184I SLM policy INITIAL has no initiator maximum for all members - ON
DEMAND class defined
DTM6422I JLS RECONCILE COMPLETE
DTM8130I SLM POLICY INITIAL VALIDATED WITH WARNING(S)
DTM8128I SLM POLICY INITIAL ACTIVE ON S02 WITH WARNINGS
DTM7525I ROBOTIC RECONCILIATION IN PROGRESS
DTM7526I ROBOTIC RECONCILIATION COMPLETE
DTM0023I TMSS INITIALIZATION COMPLETE
DTM8102I SLM SERVER FUNCTION ACTIVE ON SC74
```

Figure 6-8 Startup of Thruput Manager

JES2 sysplex for testing

Early in the project, you should consider including sandbox sysplex with JES2 as the primary subsystem that is available for all stakeholders. In the customer environment, two systems were added that run native with JES2 added. A possible merger of two JES2 systems into an existing JES3 sysplex by adding two more systems is shown in Figure 6-9.

.

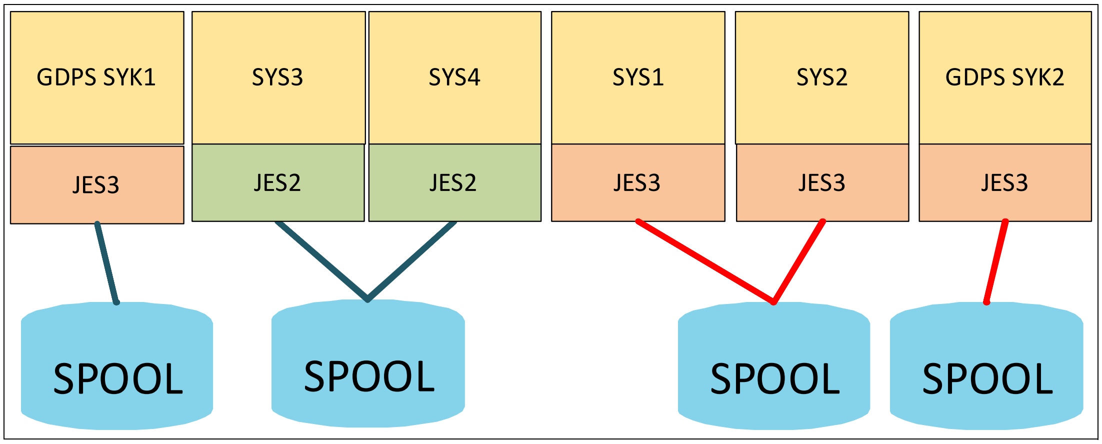


Figure 6-9 JES2 sandbox layout

If you cannot send more systems to a sysplex, consider starting JES2 as a secondary subsystem and keep JES3 as the primary, as shown in Figure 6-10.

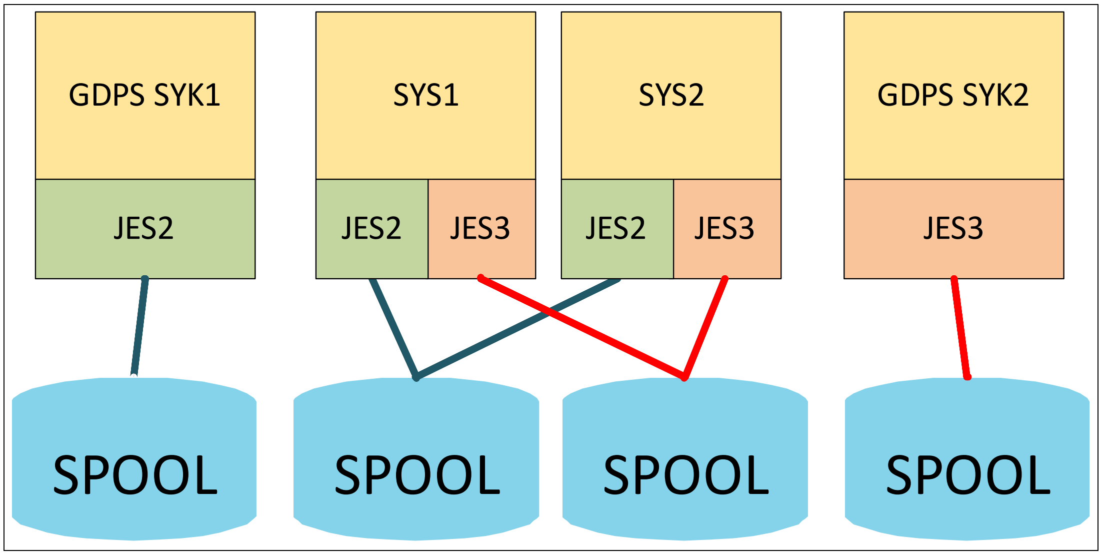


Figure 6-10 JES2 as secondary subsystem

JES2 expert nearby

During the migration project, it strongly recommended to have a JES2 expert on site or at least available by phone. Customers can experience many situations during the migration process in which access to an experienced JES2 expert was needed to quickly answer questions or solve technical problems. Otherwise, time and resources are wasted to get a problem fixed.

6.3 JES2 system design

The new JES2 system design should be flexible, easy to maintain, and simple to deploy. It also should match the following items that are compared with your existing JES3 installation:

- JES2 member names
- NJE node names
- Used Job classes
- Used Job output classes

Matching these components avoids problems after the migration process in many other areas that use fixed nodes or system names.

PARMLIB

JES2 is controlled in its configuration by a standard PARMIB member. To reduce the efforts for future maintenance, we recommend placing all common parameters in one member. All other system-specific control statements should be placed in a second PARMLIB member. If you consider operating with a JES2 printer, we recommend placing all of the printer definitions in a third PARMLIB member.

> **Hint:** If you are moving most of the JES2 configuration statements to a common PARMLIB member, use system symbols.

An example of a system-specific JES2 PARMLIB member is shown in Example 6-7. This member includes configuration statements that are unique to that particular system. All generic JES2 definitions are made available by using an INCLUDE statement in the system-specific PARMLIB member.

Example 6-7 Sample JES2 PARMLIB start member

```
/*-------------------------------------------------------------------*/
/*                                                                   */
/* GDE: JES2 INIT AND TUNING REFERENCE                               */
/* DOC: THIS MEMBER CONTAINS THE INITIAL JES2 PARMS                  */
/*                                                                   */
/* UPDATES:                                                          */
/*                                                                   */
/*-------------------------------------------------------------------*/
/* INCLUDE THE STANDARD MEMBER WITH THE GENERAL DEFAULTS             */
INCLUDE MEMBER=JES2&JESENV.STD
/*-------------------------------------------------------------------*/
/*       CHECKPOINT - CKPT1                                          */
/*                  - CKPT2 BACKUP                                   */
/*       RECONFIGURATION USING: $TCKPTDEF,RECONFIG=YES               */
/*-------------------------------------------------------------------*/
/* SEE JES2&JESENV.STD MEMBER FOR GENERAL DEFAULTS                   */
/*-------------------------------------------------------------------*/
CKPTDEF  CKPT1=(STRNAME=JES2CKPT_1,INUSE=YES)

CKPTDEF  NEWCKPT1=(DSNAME=JES2#A.&SYSNODE..&JESENV.O.CKPT1NEW,
         VOLSER=SYA411)

CKPTDEF  CKPT2=(DSNAME=JES2#A.&SYSNODE..&JESENV.O.CKPT2,
         VOLSER=SYA410,INUSE=YES)
CKPTDEF  NEWCKPT2=(DSNAME=JES2#A.&SYSNODE..&JESENV.O.CKPT2NEW,
         VOLSER=SYA411)
/*-------------------------------------------------------------------*/
CKPTSPACE BERTNUM=55100               /* BLOCK EXTENSION REUSE TABLE */
CKPTSPACE BERTWARN=70                 /* ALERT MESSAGE $HASP050      */
/*-------------------------------------------------------------------*/
/*       MAS - MULTI ACCESS SPOOL - DEFINTION                        */
/*-------------------------------------------------------------------*/
MASDEF   DORMANCY=(50,500)
MASDEF   HOLD=00000050
MASDEF   LOCKOUT=1000
/*-------------------------------------------------------------------*/
/*       MEMBER DEFINITION                                           */
/*-------------------------------------------------------------------*/
MEMBER(1) NAME=SYS1
MEMBER(2) NAME=SYS2
/*-------------------------------------------------------------------*/
/*       SPOOL VOLUMES AND DEFINITIONS                               */
/*-------------------------------------------------------------------*/
SPOOLDEF VOLUME=SYA4
SPOOLDEF DSNMASK=JES2#A.&SYSNODE..&JESENV.0.SPOOL*
/*-------------------------------------------------------------------*/
SPOOL(SYA480) DSNAME=JES2#A.&SYSNODE..&JESENV.0.SPOOL80
SPOOL(SYA481) DSNAME=JES2#A.&SYSNODE..&JESENV.0.SPOOL81
SPOOL(SYA482) DSNAME=JES2#A.&SYSNODE..&JESENV.0.SPOOL82
SPOOL(SYA483) DSNAME=JES2#A.&SYSNODE..&JESENV.0.SPOOL83
/*-------------------------------------------------------------------*/
/*       JOB DEFINITIONS                                             */
/*-------------------------------------------------------------------*/
JOBDEF   JOBNUM=40000
JOBDEF   JOBWARN=80
JOBDEF   RANGE=(1-65534)
JOBDEF   INTERPRET=INIT
JOBDEF   ACCTFLD=IGNORE
/*-------------------------------------------------------------------*/
/*       INITDEF AND INIT - DEFINE INITIATORS                        */
/*-------------------------------------------------------------------*/
INITDEF  PARTNUM=200
I(001-050) CLASS=(M1,P0,M,A),    /* 50 INIT FOR ENGINEERING+BATEMERG */
           START=YES,
           NAME=BASE
I(051-100) CLASS=(S0,S1,S2),    /* 50 INIT FOR SYSTEM JOBS           */
           START=YES,
           NAME=BSYS
I(101-200) CLASS=(S0,S1,S2,M1,P0,M), /* SPARE INIT                   */
           START=NO
/*-------------------------------------------------------------------*/
/*       OUTDEF - OUTPUT DEFAULTS                                    */
/*-------------------------------------------------------------------*/
OUTDEF   JOENUM=60000
OUTCLASS(*) BLNKTRNC=YES,       /* DEFAULTS FOR ALL CLASSES          */
         OUTDISP=(WRITE,WRITE), /* DISP NORMAL,ABEND                 */
         OUTPUT=PRINT,          /*                                   */
         TRKCELL=YES,          /*                                   */
         COMPRESS=YES           /* Enable SPOOL compression          */
OUTCLASS(C)
OUTCLASS(D)
OUTCLASS(E)
OUTCLASS(F)
OUTCLASS(G)
OUTCLASS(H)
OUTCLASS(I)
OUTCLASS(J)
OUTCLASS(L) OUTDISP=(WRITE,WRITE)
OUTCLASS(N) OUTPUT=DUMMY
OUTCLASS(O)
OUTCLASS(R) OUTDISP=(HOLD,HOLD)
/*-------------------------------------------------------------------*/
/*  INCLUDE PRINTER DEFINITIONS                                      */
/*-------------------------------------------------------------------*/
INCLUDE MEMBER=JES2PRT
/*-------------------------------------------------------------------*/
/*  DEFINE NJE NODES                                                 */
/*-------------------------------------------------------------------*/
NODE(1)  NAME=SYS1 /* OWNNODE=1                            */
NETSRV1  SOCKET=LOCAL
  SOCKET(SYS1) NODE=1,IPADDR=YOUR-ADRESS,SECURE=YES,PORT=2252
/*-------------------------------------------------------------------*/
NODE(2) NAME=SYS2
  SOCKET(SYS2) NODE=2,IPADDR=YOUR-ADRESS,SECURE=YES,PORT=2252,CONNECT=YES
NODE(3) NAME=SYS3
  SOCKET(SYS3) NODE=3,IPADDR=YOUR-ADRESS,SECURE=YES,PORT=2252,CONNECT=YES
/*-------------------------------------------------------------------*/
/*       JES2 PROCESSOR NUMBERS (TASKS)                              */
/*-------------------------------------------------------------------*/
PCEDEF   CNVTNUM=25          /* # CONVERTER          PCE'S           */
PCEDEF   OUTNUM=10           /* # OUTPUT             PCE'S           */
PCEDEF   PSONUM=10           /* # PSO                PCE'S           */
PCEDEF   PURGENUM=10         /* # PURGE              PCE'S           */
PCEDEF   SPINNUM=3           /* # SPIN               PCE'S           */
PCEDEF   STACNUM=10          /* # TSO/ STATUS/CANCEL PCE'S           */
/*-------------------------------------------------------------------*/
/*       SMF DEFINITIONS ????                                        */
/*-------------------------------------------------------------------*/
SMFDEF   BUFNUM=300          /* NUMBER OF SMF BUFFERS                */
SMFDEF   BUFWARN=80          /* WARNING THRESHOLD %                  */
/*-------------------------------------------------------------------*/
 $D INITINFO           /* WRITE INITIALIZATION INFO. TO HASPLIST     */
/*-------------------------------------------------------------------*/
 $T JOBCLASS(A),MODE=WLM /* SET CLASS=A TO WLM MANAGED ON PRODUCTION */
/*-------------------------------------------------------------------*/
```

In comparison to earlier releases of JES2, you can now use compression for any output class in your installation to lower SPOOL space. In Figure 6-9 on page 123, you see that compression is enabled for all JES2 output classes.

Information: The use of z/OS system symbols is recommended to include a more common PARMLIB member. By including this member, less effort is required later to maintain the JES2 parmlib.

The common part of our sample JES2 configuration is shown in Example 6-8. This member is valid and used for all of your systems. The following parameters can be used to code the common JES2 PARMLIB member:

PROCLIB
Use z/OS system symbols to address system-specific PROCLIB. Also, the UNCOND option allows you to define PROCLIBs that must not be available when JES2 starts. That feature allows you to define PROCLIB in the common member that is not available on all systems; for example, on IBM GDPS® controlling systems.

JOBCLASS
Define all of your job classes that your installation needs. It is recommended to lower your migration effort to define the same jobclasses you used in your JES3 installation. You can code all common parameters that apply to all job classes after the specific definitions with the JOBCLASS(*) statement. All job classes are enabled by default. If you do not want all job classes to be active, use the ACTIVE=NO option for all job classes that you do not need.

ESTLNCT
Under JES3 per default, all tasks abend with S722 when they produce more than 16 million lines of output. The ESTLNCT statement that is shown in Example 6-8 shows how to limit tasks to under 16 million lines of output (16,000 x 1000 lines of output).

NJEDEF
For performance reasons, define the number of transmit and receive paths to a maximum of four. For more information about defining NJE connections, see Chapter 6.12, “Hints and tips” on page 172.

OUTPRTY
Defines a priority of JES2 output elements based on their size. We set a common priority for all output elements as does JES3 to avoid enabling JES2 to perform a reorder for printed output.

Example 6-8 Sample common JES2 PARMLIB

```
/*--------------------------------------------------------------*/
/* GDE: JES2 INIT AND TUNING REFERENCE                          */
/* DOC: this Member contains the initial Base JES2 Parms        */
/*      included within member JES2P00 valid for all Systems.   */
/*                                                              */
/* UPDATES:                                                     */
/****************************************************************/
/*  DEFINE JES2 PROCLIB                                         */
/*--------------------------------------------------------------*/
 PROCLIB(PROC00) DD(1)=(DSN=SYS1.&SYSNODE..ZOS.PROCLIB)
 PROCLIB(PROC00) DD(2)=(DSN=SYS1.DIV.ZOS.PROCLIB)
 PROCLIB(PROC00) DD(3)=(DSN=SYS1.DIV.IBM.PROCLIB)
 PROCLIB(PROC00) DD(4)=(DSN=SYS1.&SYSNODE..SUB.PROCLIB)
 PROCLIB(PROC00) DD(5)=(DSN=PCL.U0000.P0.&SYSNODE.AKT.PROM.@008.STC),
                 UNCOND
 PROCLIB(PROC00) DD(6)=(DSN=PCL.U0000.P0.&SYSNODE.AKT.PERM.@008.STC),
                 UNCOND
 PROCLIB(PROC00) DD(7)=(DSN=JOBP.SYSA.PROC),UNCOND
 PROCLIB(PROC00) DD(8)=(DSN=JOBP.AL&RZID.A.PROC),UNCOND
/*--------------------------------------------------------------*/
/*  DEFINE JES2 CHECKPOINT                                      */
/*--------------------------------------------------------------*/
CKPTDEF  MODE=DUPLEX
CKPTDEF  DUPLEX=ON
CKPTDEF  VOLATILE=(ONECKPT=IGNORE)
CKPTDEF  OPVERIFY=NO
```

```
/*-------------------------------------------------------------------*/
/*  DEFINE JES2 MULTI ACCESS SPOOL (MAS)                             */
/*-------------------------------------------------------------------*/
MASDEF   AUTOEMEM=ON
MASDEF   OWNMEMB=&SYSNAME
MASDEF   XCFGRPNM=JES2&SYSNODE.&JESENV.   /*                         */
MASDEF   CYCLEMGT=AUTO
/*-------------------------------------------------------------------*/
/*  DEFINE JES2 SPOOL                                                */
/*-------------------------------------------------------------------*/
SPOOLDEF BUFSIZE=3992
SPOOLDEF TGSIZE=36
SPOOLDEF TRKCELL=4
SPOOLDEF LARGEDS=ALWAYS      /* MORE THAN 64 TRACKS, MAX. 1M TRACKS  */
SPOOLDEF SPOOLNUM=32
SPOOLDEF TGSPACE=(MAX=5000000)  /* 15 Mio Tracks                     */
SPOOLDEF CYL_MANAGED=ALLOWED    /*enables CYLinder managed space     */
/*-------------------------------------------------------------------*/
/*       JOB DEFINITION                                              */
/*-------------------------------------------------------------------*/
JOBDEF   PRTYJECL=NO
JOBDEF   DEF_CLASS=A
JOBDEF   CNVT_SCHENV=HONOR
JOBDEF   DUPL_JOB=DELAY
JOBDEF   ACCTFLD=IGNORE
/*-------------------------------------------------------------------*/
/*  DEFINE JES2 JOBCLASSES                                           */
/*  VARY OFF 1 SYS.(BATCH OFF): $TJOBCLASS(P0,P1,P2,P3),QAFF=-S03    */
/*  (OR SCHENV DEFAULT + DEFBASE  ???  )                             */
/*-------------------------------------------------------------------*/
                          /* SYSTEM JOBS + ENGINEERING   *************/
JOBCLASS(S0,S1,M1,M) COMMAND=IGNORE,
         MSGCLASS=T,         /* DEFAULT MESSAGE CLASS                */
         MODE=JES            /* SYST/ENG WITH JES INIT               */
JOBCLASS(S2)         COMMAND=IGNORE,
         MSGCLASS=T,         /* DEFAULT MESSAGE CLASS                */
         MODE=JES,           /* SYST/ENG WITH JES INIT               */
         XEQCOUNT=MAXI=25    /* MAXIMUM OF 25 JOBS PER SYSPLEX       */
                          /* USER JOBS                   *************/
JOBCLASS(A) COMMAND=IGNORE,
         MSGCLASS=T,         /* DEFAULT MESSAGE CLASS                */
         MODE=JES            /* WLM INIT'S                           */
                          /* PRODUCTION JOBS (BATEMERG)  *************/
JOBCLASS(P0) COMMAND=IGNORE,
         MSGCLASS=E,         /* DEFAULT MESSAGE CLASS                */
         MODE=JES            /* BATEMERG WITH JES INIT               */
                          /* PRODUCTION JOBS (BATLOW/BATMED/BATHIGH) */
JOBCLASS(P1,P2,P3) COMMAND=IGNORE,
         MSGCLASS=E,         /* DEFAULT MESSAGE CLASS                */
         MODE=WLM            /* WLM INIT'S                           */
JOBCLASS(STC,TSU) COMMAND=IGNORE,
         MSGCLASS=E          /* DEFAULT MESSAGE CLASS                */
/*-------------------------------------------------------------------*/
/* Disable all unused Jobclasses                                     */
/*-------------------------------------------------------------------*/
```

```
JOBCLASS(B,C,D,E,F,G,H,I,J,K,L,N,O,P,Q,R,S,T,U,V,W,X,Y,Z) ACTIVE=NO
JOBCLASS(0,1,2,3,4,5,6,7,8,9) ACTIVE=NO
/*-------------------------------------------------------------------*/
/* Defaults valid for all Jobclasses                                 */
/*-------------------------------------------------------------------*/
JOBCLASS(*) REGION=1200M,    /* REGION DEFAULT FÜR BATCH             */
            MSGLEVEL=(1,1),  /* JOB, ALL MSGS                        */
            SWA=ABOVE,       /* SWA CONT.BLOCKS ABOVE THE 16M-LINE   */
            PROCLIB=00,      /* PROCLIB(PROC00)                      */
            SCHENV=DEFAULT,  /* DEFAULT SCHENV  (OR DEFBASE?)        */
            TIME=(1439,00),  /* 23H + 59M TIME LIMIT FOR JOB STEP    */
            PROMO_RATE=3,    /* promotion rate for STARTBY function  */
            SYSSYM=ALLOW     /* ALLOW USAGE OF SYSTEM SYMBOLS        */
/*-------------------------------------------------------------------*/
/*       ESTBYTE - Default estimated SYSOUT bytes/Job $HASP375       */
/*-------------------------------------------------------------------*/
ESTBYTE  NUM=99999           /* 99999   KBYTES FOR 1ST MESSAGE       */
ESTBYTE  INT=50000           /*  THEN AT 50000   KBYTE INTERVALS     */
ESTBYTE  OPT=0               /* ALLOW JOBS TO CONTINUE               */
/*  0 Job is allowed to continue execution                           */
/*  1 Job is canceled without a dump                                 */
/*  2 Job is canceled with a dump                                    */
/*    (if a dump statement was coded for this job step)              */
/*-------------------------------------------------------------------*/
/*       ESTLNCT - Default estimated SYSOUT lines/Job $HASP375       */
/*-------------------------------------------------------------------*/
ESTLNCT  NUM=16000           /* Limit to 16M Lines per Job           */
ESTLNCT  INT=10000           /*  THEN AT  10K LINE INTERVALS         */
ESTLNCT  OPT=1               /* Job will be canceled by JES          */
/*-------------------------------------------------------------------*/
ESTPAGE  NUM=500             /* 1K PAGES FOR 1ST MESSAGE             */
ESTPAGE  INT=250             /*  THEN AT 100 PAGE INTERVALS          */
ESTPAGE  OPT=0               /* ALLOW JOBS TO CONTINUE               */
/*-------------------------------------------------------------------*/
/*       ESTPUN  - Default estimated PUNCH cards/Job $HASP375        */
/*-------------------------------------------------------------------*/
ESTPUN   NUM=10000           /* 10K  CARDS FOR 1ST MESSAGE           */
ESTPUN   INT=5000            /*  THEN AT 2000 CARD INTERVALS         */
ESTPUN   OPT=0               /* ALLOW JOBS TO CONTINUE               */
/*-------------------------------------------------------------------*/
/*       INTRDR - Internal Reader Definition                         */
/*-------------------------------------------------------------------*/
INTRDR   CLASS=A             /* DEFAULT JOB CLASS                    */
/*-------------------------------------------------------------------*/
/*       LOADMOD/EXIT - JES2 EXITS                                   */
/*-------------------------------------------------------------------*/
LOADMOD(HASX06A)             /* EXIT FOR DD DSN=...#DT#              */
EXIT(6)  ROUTINES=(EXIT06)   /* JES2 CONVERTER EXIT                  */
LOADMOD(HASX23A) STORAGE=CSA /* EXIT FOR PSF HEADER,TRAILER          */
EXIT(23) ROUTINES=(EXIT23)   /* JES2 FSS EXIT                        */
/*-------------------------------------------------------------------*/
/*       CONDEF - Console Definition                                 */
/*-------------------------------------------------------------------*/
CONDEF   DISPLEN=70          /* LENGTH OF OUTPUT LINES ON CONSOLE    */
CONDEF   DISPMAX=1000        /* # OF OUTPUT LINES/MSG  ON CONSOLE    */
CONDEF   CMDNUM=3000         /* CONSOLE MESSAGE BUFFER               */
CONDEF   BUFNUM=3000         /* CONSOLE MESSAGE BUFFER               */
CONDEF   BUFWARN=50          /* Warnings at 50% usage                */
/*-------------------------------------------------------------------*/
/* NJE Definitions                                                   */
/*-------------------------------------------------------------------*/
NJEDEF  LINENUM=15           /* MAX NUMBER OF NJE LINE'S             */
NJEDEF  JRNUM=4              /* MAX NUMBER OF Job receiver           */
NJEDEF  JTNUM=4              /* MAX NUMBER OF Job transmitter        */
NJEDEF  SRNUM=4              /* MAX NUMBER OF Sysout Receiver        */
NJEDEF  STNUM=4              /* MAX NUMBER OF Sysout Transmitter     */
NJEDEF  NODENUM=15           /* MAX NODE NUMBER                      */
NJEDEF  OWNNODE=1            /* NODE(1)                              */
NJEDEF  TIMETOL=0            /* ALLOW TIME DIFFERENCES with PTA      */
NODE(*) PATHMGR=NO           /* PATHMGR not used due to conn failures*/
LINE(1-15) UNIT=TCP,START=YES,CONNECT=(YES,10)
/*-------------------------------------------------------------------*/
/* PCE Definitions                                                   */
/*-------------------------------------------------------------------*/
SUBTDEF  GSUBNUM=40          /* # of JES2 worker tasks               */
PCEDEF   CNVTNUM=25          /* # CONVERTER          PCE'S           */
PCEDEF   OUTNUM=25           /* # OUTPUT             PCE'S           */
PCEDEF   PSONUM=10           /* # PSO                PCE'S           */
PCEDEF   PURGENUM=25         /* # PURGE              PCE'S           */
PCEDEF   SPINNUM=10          /* # SPIN               PCE'S           */
PCEDEF   STACNUM=10          /* # TSO/ STATUS/CANCEL PCE'S           */
/*-------------------------------------------------------------------*/
/* Set Output Priority default to 8                                  */
/*-------------------------------------------------------------------*/
OUTPRTY(*) PAGE=16000000,PRIORITY=8,RECORD=16000000
```

PROCLIB

The guidelines that we applied to the PARMLIB can be used for the PROCLIB definition. The JES2 start procedure was made flexible to address many needs. An example is shown in Example 6-9.

Example 6-9 Sample of JES2 start procedure

```
//JES2    PROC P=WARM,R=NOREQ,
//        M=00,              00 = ACTIVE, 90=BACKUP
//        JE=&JESENV,        P = PROD-JES, T=TEST-JES
//        JO='DSORG=PS',     DUMMY = SET HASPLIST TO DUMMY
//        RZ=&RZDSN          DEFAULT FROM SYSTEM SYSTEM-SYMBOL
//*******************************************************************
//*  FUNCTION:      START JES2 SUBSYSTEM
//*  RESPONSIBLE:   z/OS Department
//*  AT ABEND:      CALL MVS ON CALL SERVICE
//*
//*     START JES: /S JES2
//*      STOP JES: /$PJES2
//*      STOP JES: /$PJES2,ABEND
//*
//*******************************************************************
//ALLOC   EXEC PGM=IEFBR14,TIME=1440
//DD1       DD &JO,
//   DSN=JES2#A.&RZ..&JE.O.&SYSNAME..HASPLIST.D&LYYMMDD..T&LHHMMSS,
//             UNIT=DISK,MGMTCLAS=COM#A064,
//             SPACE=(TRK,(15,15),RLSE),DISP=(NEW,CATLG),
//             LRECL=121,RECFM=FBA
//*
//JES2    EXEC PGM=HASJES20,TIME=1440,
//    PARM='&P.,PARMLIB_MEMBER=JES2&JE.&M.,&R'
//STEPLIB   DD DISP=SHR,DSN=SYS1.SHASLNKE
//HASPLIST  DD &JO,
//             DSN=*.ALLOC.DD1,DISP=SHR
//*
```

The start parameters are listed in Table 6-2.

Table 6-2 JES2 start parameters

Start parameter	Description
P	You specify the wanted start mode of JES2. The default is a JES2 warm start if no parameter is passed. P=COLD proceeds a JES2 cold start.
R	This option allows you to start JES2 automatically without showing you a $HASP400 ENTER REQUESTS message. If pass R=REQ to the start procedure, JES2 prompts the message and waits for replies.
M	With this parameter, the type of primary JES2 parmlib member that should be used for this start can be controlled. In our example, you pass a suffix to the procedure that concatenated the JES2 parameter member that is called JES2Pxx to the final member name.
JE	This parameter controls the type of JES2 you want to use. In this case study, we used two different configurations: One for test purposes and the other for production. This parameter also can be controlled by an external system symbol.
JO	With this parameter, you can activate a type of logging of all parameters that are passed to JES2 during start and store them in a separate data set or GDG.
RZ	This parameter can be used for allocating system-specific data sets (logs) during start.

The parameters that are listed in Table 6-2 are recommendations to demonstrate the flexibility of JES2. They can be changed or extended based on customers needs.

Initiators

The calculation for the new JES2 initiators is based on the system layout under JES3.

Attention: If you plan the JES3 migration from a version of z/OS before V2R1, consider upgrading your system to at least z/OS V2R1 level first. With this release, JES2 supports eight-character job classes instead of one-character as in releases that are older than V2R1.

In Example 6-7 on page 124, we define 200 initiators per system by default. This amount is much higher than needed. This high number of initiators is used so that spare initiators can be defined in case more are needed. As shown in Example 6-7 on page 124, initiators 101 - 200 are defined with the START=NO option.

The grouping of initiators to job classes should be planned and depends on the customers environment.

Checkpoint

A single resource is available in JES2 that is called CHECKPOINT. This resource is used to share relevant JES2 information across all participating JES2 MAS members in the sysplex. Each of the participating JES2 members has a dedicated, defined time to access the checkpoint.

> **Attention:** The JES2 checkpoint is a sensitive resource and requires careful handling. If it is not set properly, it can cause serious problems later. Also, we recommend placing the backup for CKPT1 on DASD instead of using an alternative CF structure.

Consider the following recommendations for the JES2 checkpoint:

- Place CKPT1 in the Coupling Facility (CF), if possible.
- Place CKPT2 on a separate DASD; no other data set should be allocated on that DASD.

Have a backup for CKPT1 and CKPT2 available. For safety reasons, the backup for both CKPT1 and CKPT2 should be allocated on separate DASD volumes other than the primary CKPT1 and CKPT2. The resulting checkpoint definition that is based on the configuration is shown in Example 6-10.

Example 6-10 Display of active Checkpoint definitions

```
$DCKPTDEF
$HASP829 CKPTDEF
$HASP829 CKPTDEF  CKPT1=(STRNAME=JES2CKPT_1,INUSE=YES,VOLATILE=YES),
$HASP829          CKPT2=(DSNAME=JES2#A.RZ4.PO.CKPT2,VOLSER=SYA410,
$HASP829          INUSE=YES,VOLATILE=NO),

$HASP829          NEWCKPT1=(DSNAME=JES2#A.RZ4.PO.CKPT1NEW,
$HASP829          VOLSER=SYA412),
$HASP829          NEWCKPT2=(DSNAME=JES2#A.RZ4.PO.CKPT2NEW,
$HASP829          VOLSER=SYA411),MODE=DUPLEX,DUPLEX=ON,LOGSIZE=1,
$HASP829          VERSIONS=(STATUS=ACTIVE,NUMBER=50,WARN=80,MAXFAIL=0,
$HASP829          NUMFAIL=0,VERSFREE=50,MAXUSED=2),RECONFIG=NO,
$HASP829          VOLATILE=(ONECKPT=IGNORE,ALLCKPT=WTOR),OPVERIFY=NO
```

SPOOL

The JES2 SPOOL should be placed on dedicated volumes to avoid any ENQ or RESERVES from others to that data sets or DASD. The SPOOL size depends on your environment and should also include a reserve space. In our experience, the SPOOL space usage between JES2 and JES3 is approximately the same; however, under JES2, we used double the SPOOL space size as we did under JES3. The SPOOL values are compared in Table 6-3.

Table 6-3 Comparison of SPOOL values

Value	Total cylinder JES3	Total cylinder JES2
SPOOL size	225175 Cyl	480640 Cyl
Number of DASD	31	8

Under JES3, the average SPOOL utilization was approximately 50%. Under JES2, we see an average of 29% SPOOL utilization, as shown in Example 6-11.

Example 6-11 Sample JES2 SPOOL utilization

```
$DSPL
$HASP893 VOLUME(SYA280)  STATUS=ACTIVE,PERCENT=29
$HASP893 VOLUME(SYA281)  STATUS=ACTIVE,PERCENT=29
$HASP893 VOLUME(SYA282)  STATUS=ACTIVE,PERCENT=29
$HASP893 VOLUME(SYA283)  STATUS=ACTIVE,PERCENT=29
$HASP893 VOLUME(SYA284)  STATUS=ACTIVE,PERCENT=29
$HASP893 VOLUME(SYA285)  STATUS=ACTIVE,PERCENT=29
$HASP893 VOLUME(SYA286)  STATUS=ACTIVE,PERCENT=29
$HASP893 VOLUME(SYA287)  STATUS=ACTIVE,PERCENT=29
$HASP646 29.2025 PERCENT SPOOL UTILIZATION
```

Note: We did not see any performance issue when the number of SPOOL volumes was decreased compared to the JES3 setup.

The SPOOL definitions are set according to the IBM recommendations, as shown in Example 6-12.

Example 6-12 Sample JES2 SPOOL definitions

```
$DSPOOLDEF
$HASP844 SPOOLDEF
$HASP844 SPOOLDEF  BUFSIZE=3992,DSNAME=SYS1.HASPACE,
$HASP844           DSNMASK=JES2#A.RZ2.PO.SPOOL*,FENCE=(ACTIVE=NO,
$HASP844           VOLUMES=1),GCRATE=NORMAL,
$HASP844           LASTSVAL=(2015.290,16:06:08),LARGEDS=ALWAYS,
$HASP844           SPOOLNUM=32,TGSIZE=36,TGSPACE=(MAX=5000416,
$HASP844           DEFINED=2403340,ACTIVE=2403340,PERCENT=29.2117,
$HASP844           FREE=1701283,WARN=80),TRKCELL=4,VOLUME=SYA2
```

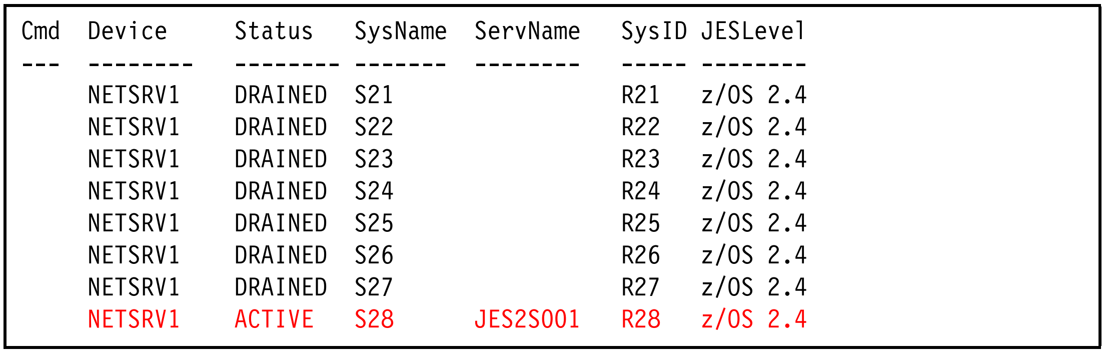
NJE

The NJE setup was copied from the previous JES3 setup. All node names remain the same to be compatible with all your applications that use node names. No other actions are needed here, except if your are planning to transfer spool data sets during migration from JES3 to JES2.

Attention: In comparison to JES3, every JES2 instance features its own NJE NETSRV. Therefore, more than one NETSRV is active in a sysplex at the same time. This configuration caused misleading messages to appear on the console.

It is recommended to control the JES2 NETSRV with your system automation. You must be sure that only one NETSRV server is active at the same time in the sysplex. The system automation should control the JES2 NETSRV by using the `$SNET` and `$SNETSRV1` commands.

An example of how NJE servers should be running is shown in Figure 6-11 on page 134.



```
Cmd  Device      Status    SysName  ServName    SysID JESLevel
---  --------    -------- -------   --------    ----- --------
     NETSRV1     DRAINED   S21                  R21   z/OS 2.4
     NETSRV1     DRAINED   S22                  R22   z/OS 2.4
     NETSRV1     DRAINED   S23                  R23   z/OS 2.4
     NETSRV1     DRAINED   S24                  R24   z/OS 2.4
     NETSRV1     DRAINED   S25                  R25   z/OS 2.4
     NETSRV1     DRAINED   S26                  R26   z/OS 2.4
     NETSRV1     DRAINED   S27                  R27   z/OS 2.4
     NETSRV1     ACTIVE    S28      JES2S001    R28   z/OS 2.4
```

Figure 6-11   NETSRV sysplex view

JES2 cold start

After all of the tasks that are described thus far in this chapter are completed successfully, initialize your checkpoint data sets and your SPOOL data sets. This process can be done by using JES2 cold start.

With this cold start, all checkpoint data sets and your SPOOL are initialized and cleared. This task can be done under JES3 as primary subsystem by configuring JES2 as a secondary subsystem in your system. An example of a JES2 cold start is shown in Example 6-13.

Example 6-13   Sample JES2 cold start

```
S JES2,P=COLD
IEF403I JES2 - STARTED - TIME=20.33.48     S00
 IEE252I MEMBER  JES2T00 FOUND IN SYS1.RZO.ZOS.PARMLIB
 IEE252I MEMBER  JES2T00 FOUND IN SYS1.RZO.ZOS.PARMLIB
 IEE252I MEMBER JES2PSTD FOUND IN SYS1.DIV.ZOS.PARMLIB
 IEE252I MEMBER JES2PSTD FOUND IN SYS1.DIV.ZOS.PARMLIB
 IXZ0001I CONNECTION TO JESXCF COMPONENT ESTABLISHED, 382
        GROUP JES2RZOT MEMBER RZOT$S00
 IEF403I IEESYSAS - STARTED - TIME=20.33.49  S00
 $HASP9084 JES2 MONITOR ADDRESS SPACE STARTED FOR JES2
 $HASP537 THE CURRENT CHECKPOINT USES 5282 4K RECORDS
*$HASP436 CONFIRM z22 MODE COLD START ON 417
  CKPT1 - VOLSER=SYA013 DSN=JES2#A.RZO.TO.CKPT1
  CKPT2 - VOLSER=SYA013 DSN=JES2#A.RZO.TO.CKPT2
  SPOOL - PREFIX=SYA01  DSN=SYS1.HASPACE
*374 $HASP441 REPLY 'Y' TO CONTINUE INITIALIZATION OR 'N' TO TERMINATE IN RESPONSE
TO MESSAGE HASP436
R 374,Y
 IEE600I REPLY TO 374 IS;Y
 $HASP478 INITIAL CHECKPOINT READ IS FROM CKPT1 424
          (JES2#A.RZO.TO.CKPT1 ON SYA013)
          LAST WRITTEN TUESDAY, 19 JUN 2018 AT 18:28:52 (GMT)
*$HASP493 JES2 COLD START IS IN PROGRESS - z22 MODE
 $HASP266 JES2 CKPT2 DATA SET IS BEING FORMATTED
 $HASP267 JES2 CKPT2 DATA SET HAS BEEN SUCCESSFULLY FORMATTED
 $HASP266 JES2 CKPT1 DATA SET IS BEING FORMATTED
 $HASP267 JES2 CKPT1 DATA SET HAS BEEN SUCCESSFULLY FORMATTED
 $HASP850 5000 TRACK GROUPS ON SYA013
 $HASP851  4995416 TOTAL TRACK GROUPS MAY BE ADDED
```

```
IXZ0001I CONNECTION TO JESXCF COMPONENT ESTABLISHED, 435
         GROUP SYSJ2$XD MEMBER JES2RZOT$S00$$$$
$HASP492 JES2 COLD START HAS COMPLETED - z22 MODE
$HASP261 Member S00 performs deadline scheduling processing
$HASP249 COMMAND RECEIVED FROM INITIALIZATION 438
$DINITINFO
$HASP825 INITINFO 439
$HASP825 INITINFO  --- Command used to start JES2
$HASP825           S JES2,P=COLD
$HASP825           --- HASPPARM data sets read
```

Information: In customers' environments, a JES2 cold start with eight 3390-54 volumes that are defined for SPOOL took almost 9 minutes to complete.

6.4 Educating stakeholders

One of the most important aspects of migration is a working communication concept and in parallel, supportive education for stakeholders. These facets increase the acceptance of the project significantly.

It is suggested to provide a general education session for stakeholders that can include the following topics:

- JES2 overview: JES3 differences:
  - JES2 overview: Multi-access Spool
  - JES3 overview: Complex
  - JES2: Data sets
  - JES3: Data sets
  - JES2: Control
  - JES3: Control
- JES2 start and control:
  - Multi-access Spool: XCF
  - JES2 initialization
  - JES2 procedure
  - JES2 control START: COLD
  - JES2 control START: WARM
  - JES2 warm start types
  - JES2 address spaces
  - JES2 control stop
  - Poly/Secondary JES2
  - JES2 functions: JES3 migration
  - What JES2 does
  - What JES3 does
  - JES2: JES3 differences
- JES2: A job life:
  - Job phases
  - Input: Submit
  - Conversion
  - Conversion and interpretation
  - Waiting on conversion: TYPRUN=JCLHOLD
  - JCL error: TYPRUN=HOLD

  - TYPRUN=HOLD
  - Scheduling: JES
  - Scheduling: WLM
  - Queue affinity: JOBCLASS QAFF
  - Queue hold: JOBCLASS QHELD
  - Execution count: JOBCLASS XEQC
  - Stop Member Execution – $PXEQ
  - Stop JES2 Member: $P
  - JES initiator status
  - JES initiator halted or drained: $ZI/$PI
  - All JES initiators in use
  - All WLM initiators in use
  - Duplicate job name
  - Duplicate job name: Support on JOBCLASS
  - Job waiting for scheduling environment
  - Start job $SJ
  - Execution
  - Data set handling: JES3
  - Data set handling: JES2 (no SETUP or Locate)
  - Waiting for data set
  - Multiple jobs or data sets waiting
  - Requeue job: Waiting for data set
  - Send a message to a job log: EXEC
  - Output class
  - Output disposition
  - OUTPUT statement
  - Output group
  - Output group: JES2 display
  - Output group: SDSF display
  - Output samples

- JES2 features:
  - System/Sysaff (JES3 2.1)
  - JOB CLASS support (JES3 2.1)
  - Job correlator: UJOBCORR
  - Spin off JESLOG (JES3)
  - Spin off SYSOUT
  - Return and condition code step and job
  - Instream PROCs and INCLUDEs (JES3 2.1)
  - System symbols in batch (JES3 2.1)
  - Instream symbols
  - Big step program parameters (JES3 2.1)
  - Batch job enqueue downgrade
  - TSO multiple logon (JES3 2.1)

- JES2 JCL migration:
  - JCL changes: JOB
  - JCL changes: DD
  - JCL changes: OUTPUT
  - JCL changes: others
  - JES3 JECL statements
  - JES2 JECL statements

- JES2 commands overview:
  - JES2 commands in general
  - Display job: $DJ

  - Set and modify job: $TJ
  - Spool monitoring
  - Other shortages

The basic education session is held at the beginning of the migration project and often is repeated shortly before the migration starts. As the same time, the following special education sessions can be added for your subject matter experts:

- System Engineering z/OS:
  - How JES2 works
  - Dealing with SPOOL and checkpoint
  - Checkpoint reconfiguration
  - How to do performance analyses
  - Restart and recovery
- Operators:
  - JES2 startup and shutdown
  - Most important JES2 commands for operation
- Print operators:
  - Most important JES2 commands for printing
  - Dealing with printers
  - Managing JES2 output

6.5 Removing and replacing JES3 exits

The removal or replacement of all installed JES3 user exits to JES2 depends on the customer's installation. During the migration project, we created a list of all installed JES3 exits and had to make the following decisions:

- Keep the exits and transfer them to the appropriate JES2 exit.
- Delete the exit that is under JES3.
- No longer needed under JES2, so leave them as is alone and delete them during migration.

Important: All newly created JES2 exits *must* be on the same software level, such as the destination JES2 level.

Next, we describe some general recommendations for migrating JES exits. The use of JES2 exits is shown in Figure 6-12.

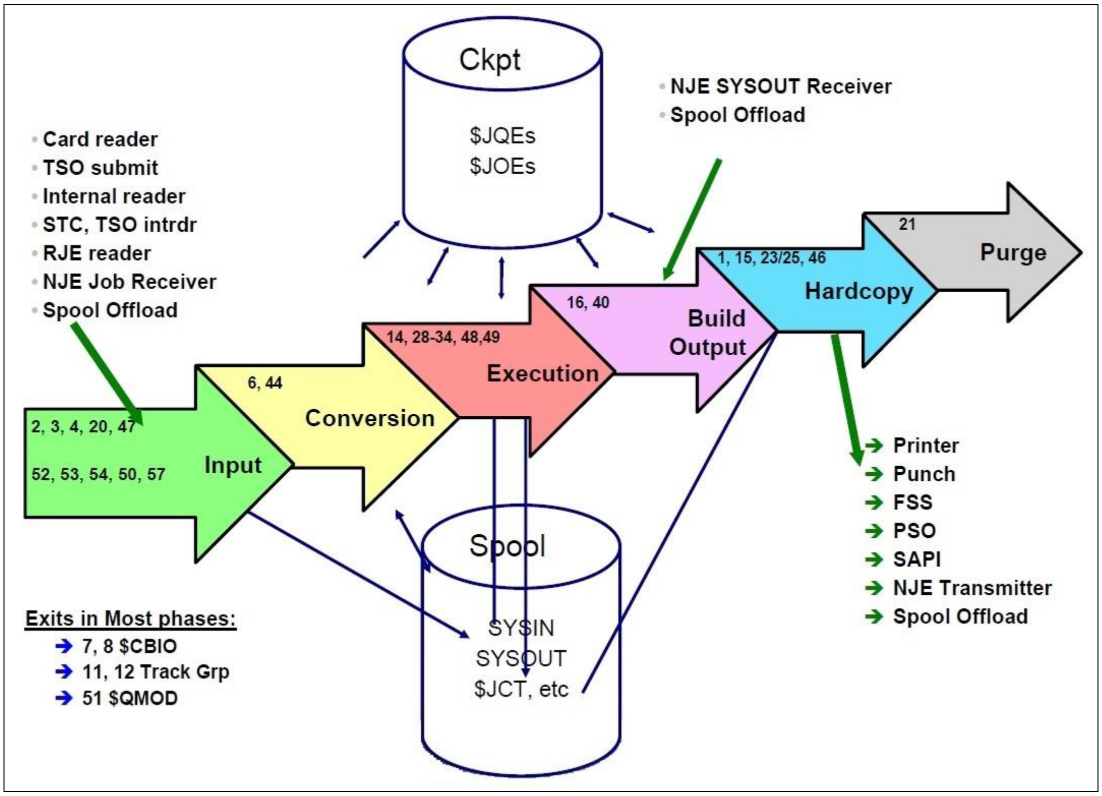


Figure 6-12 JES2 exit flowchart

The execution environments for JES2 exits are listed in Table 6-4.

Table 6-4 JES2 exit execution environments

Type	Location	Description
1	JES2 Main task	Included in the module HASJES20. It is loaded into a private area of JES2 and run under the control of JES2 (in HASPNUC). Use the JES2 macro $WAIT instead of the MVS WAIT macro. JES2 Dispatcher controls all processing within the main task environment. MVS WAITs only in JES2 exits 0, 19, 24, and 26 (according to the IBM manual).
2	JES2 Sub task	Run in the private area of JES2 address space but run asynchronously with the JES2 main task, WAIT, and POST operation and system-wide MVS Services are available. Many JES2 main task data areas are directly addressable, but users of these resources must understand when and where serialization of these resources is relevant.

Type	Location	Description
3	User Environment	In common storage and run in the users address space. System-wide MVS Services are available. The environment is more complex and includes many integrity, synchronization, locking, and cross-address space communications considerations. Special operating environment called (USER,ANY) - ENVIRON=(USER,ANY) on $MODULE or $ENVIRON statement (R11 = HCCT address). If the routine is called by the JES2 main task, $SAVE/$RETURN are called. It is not possible to work with the linkage stack (BAKR). In any environment, a PSV-type save area is obtained rather than using a BAKR.
4	FSS Address space	Functional subsystem (FSS) is in the functional subsystem address space. Similar to the user environment (JES2 services are limited). Task interaction within the FSS. All data areas and control blocks are not accessible from the FSS. Accessible control blocks are $JOE, $JIB, FSSCB, FSACB, and system-wide MVS services.

JES2 can install or activate a maximum of 256 exits. EXIT 1 - EXIT 60 are provided by IBM, although the samples do not always correspond to what is expected as good examples.

The exits have their own macros (usually $xxx) and these macros can cause problems when used with MVS macros. In exit 6, we wanted to use the Macro TCB. A collision occurred because parts of the TCB macro are used in the JES2 macros, which did not display the HLASM during the assembly. Instead, it stopped the assembly without writing out any warning.

The JES2 control blocks, which are described in the JES2 Data Areas manuals, are a good way to access important data. In the JES2 exits, for example, the JOBNAME can be found by using the field JCTJNAME, which completely replaces the variant to be run by using MVS control blocks.

The exits can also change data in the control blocks or pass data on to the following exits; for example, JCTXMASK in the JCT, which can be changed by each exit (for example, EXIT 52). We attempted to avoid a GETMAIN or STORAGE OBTAIN in all our previously programmed exits because this configuration seemed to us to be a considerable burden for the systems with the expected exit calls.

Because the exits need many registers (GPR or GR) for addressing JES2 control blocks, one should have more than 16 registers available for programming.

In programs that do not run in AMODE 64, the right part of the registers (bit 32 - 63) can be moved to the left part (bit 0 - 31) with the OP instructions SLLG or SRLG to save the registers. We used this kind of register storage several times in exits 52, 54, and 6.

Attention: Because exits are also used with the TSO Logon, it might be impossible to perform a logon in the TSO in certain situations.

For testing purposes, you can use some of the JES2 commands that are listed in Table 6-5 to dynamically activate or deactivate JES2 exits.

Table 6-5 Useful JES2 exit commands

Command	Description
$ADD LOADMOD(x),STOT=PVT	Load a load module
$DEL LOADMOD(x)	Delete a load module
$T LOADMOD(x),REFRESH	Reload a new copy of a load module
$T EXIT(n),ROUTINES=	Change routines in list ROUTINES=+routine or ROUTINES=-routine allowed
$T EXIT(n),REFRESH	Locate most recent copy of exit routine
$D EXIT(n),LONG	Display more information
$T EXITt(006),STATUS=DISABLED	Disable (deactivate) an activated JES2 exit on that particular system
$TEXIT(xx),ROUTINE=<your mod>,STATUS=ENABLED,TRACE=NO	Enable (activate) a loaded JES2 exit on that particular system

When programming the exits, they almost always are called in supervisor status and run in key 0 or key 1, depending on the environment. Key 0 and supervisor status can lead to problems if errors occur (some common storage areas are overwritten unintentionally).

Hint: The exits should be defined as a user modification so that they can be imported by way of SMP/E.

Customer-installed JES3 exits are listed in Table 6-6.

Table 6-6 Customers JES3 exits

JES3 Exit	Description	Action
IATUX69	LOCAL MESSAGE EXIT	Removed; custom made DEADLINE processing replaced
IATUX70	GLOBAL MESSAGE EXIT	Removed; custom made DEADLINE processing replaced
IATUD02	DSP LOCATE	No carry over; function does not exit under JES2
IATUD05	DSP II/ INQUIRE INITS	No carry over; function does not exit under JES2
IATUX09	INTRDR POSTSCAN EXIT	To be defined in JES2 PARMLIB
IATUX03	MODIFY JCL CHANGE ADD DSNAME	Removed; application that includes needed modified JES3 message is changed
IATUX29	CHANGES IN IATUX29 SET SA-LIM TO 200 FOR IBM IMS - JOBS (CNTL-,MPP-) NO PROD-CLS FOR TESTJOBS	Removed because SETUP no longer exits
IATDLTM	DEADLINE, ISSUE AUTOM. F J/NNNN R	Removed; custom made DEADLINE processing replaced

JES3 Exit	Description	Action
IATUD08	DSP INTERC	Removed; no longer used
IATUX19	Printer output routing selection	Removed
IATUX04	CHECK FOR CORRECT PROKOS-# IN JOB ACCOUNTING	Removed; in JES2, we use another method to pass parameter for start
IATUX15	INISH - DECK MODIFICATION EXIT	Removed; custom made function that can be removed
IATGRPT	FUNCTION CONTROL TABLE DEFINE USER DSP'S	Removed; no longer needed with JES2
IATOSFP	MSG-CHANGE IAT7007	Removed; belongs to IATUX03 and application was changed
IATUX28 IATUX29 IATUX33 IATUX40 IATSIOR	ACF2 interceptor exits	Use of JES2 interceptors instead (for more information, see Example)
IATUX45	3800-3 IATUX45	Transfer the function to JES2 exit HASX23A

Many of these exits belong to JES3 unique functions and do not feature anything to migrate (see Example 6-16).

Example 6-14 Sample ACF2 exit interceptors

```
LOAD(ACFJ2ITF) STOR=CSA                       /* ACF2/JES2 interface */
EXIT2   ROUTINE=ACFEXIT2                    /* job card scan routine */
EXIT4   ROUTINE=ACFEXIT4                    /* jcl card scan routine */
EXIT20  ROUTINE=ACFEXT20                       /* end-of-rdr manager */
EXIT24  ROUTINE=ACFEXT24                 /* Post-Initialization Exit */
EXIT26  ROUTINE=ACFEXT26                         /* Termination Exit */
EXIT31  ROUTINE=ACFEXT31             /* SSI Data Set Allocation Exit */
EXIT34  ROUTINE=ACFEXT34           /* SSI Data Set Unallocation Exit */
EXIT46  ROUTINE=ACFEXT46                        /* NJE Transmit Exit */
EXIT50  ROUTINE=ACFEXT50    /* end-of-rdr manager - user environment */
EXIT52  ROUTINE=ACFEXT52 /* job card scan routine - user environment */
EXIT54  ROUTINE=ACFEXT54 /* jcl card scan routine - user environment */
EXIT56  ROUTINE=ACFEXT56     /* NJE Transmit Exit - user environment */
EXIT225 ROUTINE=ACFEX225                 /* subtask attach/post rtne */
EXIT227 ROUTINE=ACFEX227,DISABLE            /* debug message routine */
```

Using JES2 policies

Another possibility to migrate JES3 exits is the new function that is called *JES2 policies*. JES2 introduces JES2 policies as a way to customize JES2 processing without exit programming. This support also reduces the need for specific JES2 logic skills and improves JES2 reliability by isolating JES2 from bugs in JES2 exit programs.

JES2 policies provide a way to customize JES2 processing without programming. Users formulate customization requirements in high-level terms based on general understanding of z/OS jobs and their attributes. These requirements are defined to the JES2 in a high-level human readable syntax. JES2 code applies customization requirements that are described in applicable policies in strategic points in JES2 processing.

> **Attention:** JSON names are case-sensitive and must be entered exactly as defined.

An example of the use of JES2 policies is shown in Figure 6-13.

```
{  "policyName":     "CONVPOL11",
      "policyVersion":  1,
      "policyType":     " JobConversion ",
      "definitions":
         [
           { "condition" : " JobHasAffinity('SC74')  ",
              "actions"  :
                 [
                    {  "action"    : " modifyJob ",
                       "attribute" : " SYSAFF ",
                       "value"     : " listAdd(sysaff, 'SC75') "
                    },
                    {  "action"  : " SendMessage ",
                       "message" : " 'New job affinity is ' || string(SYSAFF) "
                    }
                 ]
           }
         ]
   }
```

Figure 6-13 Sample JES2 policy

After importing and activating these policies into JES2, all jobs that are running through the converter are checked for system affinity to system SC74 (in this case).

If a system affinity to system SC74 is detected, the incoming job is modified in terms of changing the system affinity to system SC75. A separate message also is displayed that informs the customer that this particular job was modified by JES2.

At the time of this writing, only policytype JobConversion is available for use.

Print exits

In a customer environment, PSF is used to print documents on high-end printers and office printers. Some documents required a Header and Trailer page. Therefore, several PSF exits are in place, as listed in Table 6-7.

Table 6-7 Sample JES Printer exits

JES3 Exit	JES2 Exit	Description
IATUX45	HASX23A	Exit is needed to pass JES2 information to the FSS routine.
APSUX01	APSUX01	PSF Header exit that must be created according to your requirements.
APSUX02	APSUX02	PSF Trailer JES2 exit that must be created according to your requirements.
APSUX03	APSUX03	PSF Header JES2 exit that must be created according to your requirements.
APSUX06	APSUX06	PSF Message exit that must be created if you must change PSF messages before they are sent to console.

Job and user information often is used on header pages.

6.6 Transforming JES3 special functions

JES3 provides some special functions that are not available in JES2 or for which a full equivalent in JES2 does not exist.

Information: Most of these functions, such as DJC and MDS, can be disabled before the JES3 migration is started. Doing so makes the migration more agile and less prone to error.

Dependent job control

Dependent job control (DJC) is a method of handling multiple jobs that must be run in a specific order because of job dependencies. DJC manages jobs that depend on one another.

Success or failure of one job can result in execution, holding, or cancellation of other jobs. This function is intended to implement some dependencies while running jobs. If possible, all of those jobs should be moved in your professional BATCH scheduling system before migrating JES3.

To identify those job candidates in your system, we suggest scanning your OPERLOG or DLOG for the last year for such messages, as shown in the following example:

```
IAT6160 JOB NET xxxx NOW ENTERING SYSTEM
IAT6100 (JOB25676) JOB xxx (JOBxxxxx),PRTY=01,ID=LUTZ NET-ID=xx SUB=JOB25494
```

The professional BATCH scheduling system is sometimes not useful or flexible, especially for engineering jobs. Since z/OS V2R2, the user can use the new JES2 job group function (see Example 6-15 on page 144).

Example 6-15 Example of using JES2 JOBGROUP

```
//LUTZ     JOBGROUP (SYSPRG,MIST,,0815),
//            'KÜHNER',
//            OWNER=A710622,
//            HOLD=NO,
//            ONERROR=(STOP),
//            SYSTEM=(SC80),
//            SCHENV=DEFAULT
//A710JOBA GJOB
//A710JOBB GJOB
//         AFTER NAME=A710JOBA,WHEN=(RC=0)
//A710JOBC GJOB
//         AFTER NAME=A710JOBA,WHEN=(RC=4)
//A710JOBD GJOB
//         AFTER NAME=A710JOBB
//         AFTER NAME=A710JOBC
//LUTZ     ENDGROUP
//A710JOBA JOB (RJ089350,LIPA,,1157),MSGCLASS=T,MSGLEVEL=(1,1),
//    CLASS=M,NOTIFY=&SYSUID
// SCHEDULE JOBGROUP=LUTZ
//IEFBR1   EXEC PGM=IEFBR14
//
//A710JOBB JOB (RJ089350,LIPA,,1157),MSGCLASS=T,MSGLEVEL=(1,1),
//    CLASS=M,NOTIFY=&SYSUID
// SCHEDULE JOBGROUP=LUTZ
//IEFBR1   EXEC PGM=IEFBR14
//
//A710JOBC JOB (RJ089350,LIPA,,1157),MSGCLASS=T,MSGLEVEL=(1,1),
//    CLASS=M,NOTIFY=&SYSUID
// SCHEDULE JOBGROUP=LUTZ
//IEFBR1   EXEC PGM=IEFBR14
//
//A710JOBD JOB (RJ089350,LIPA,,1157),MSGCLASS=T,MSGLEVEL=(1,1),
//    CLASS=M,NOTIFY=&SYSUID
// SCHEDULE JOBGROUP=LUTZ
//IEFBR1   EXEC PGM=IEFBR14
//
```

As shown in Example 6-15, four jobs that are defined in a JES2 job group that is named LUTZ. Each of the participating jobs in that job group must be defined by a JCL GJOB statement and (optionally) a condition.

After submitting the group of jobs, you should see the successfully registered messages from JES2, as shown in Example 6-16.

Example 6-16 Successfully job group registration

```
11.45.17 JOB02568  $HASP1300 A710JOBA registered to job group LUTZ
11.45.17 JOB02568  $HASP1301 A710JOBA in job group LUTZ queued for execution
11.45.17 JOB02569  $HASP1300 A710JOBB registered to job group LUTZ
11.45.17 JOB02570  $HASP1300 A710JOBC registered to job group LUTZ
11.45.17 JOB02571  $HASP1300 A710JOBD registered to job group LUTZ
```

Based on Example 6-15, we show you the same logic in Figure 6-14 on page 145.

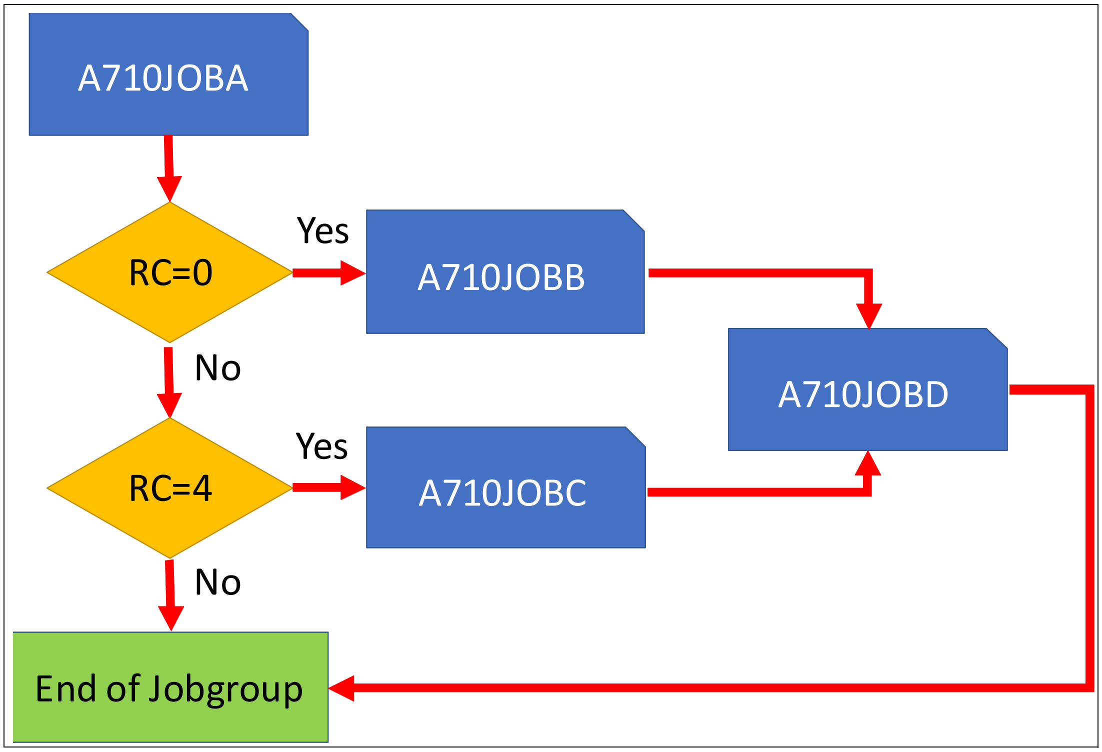


Figure 6-14 Visual example of the job flow

Upon completion, you should see messages similar to the example that is shown in Example 6-17. This example shows the start of job A710JOBA and its end with RC=0000. This issue caused two results: First, the A710JOBC is canceled because of the mismatch of the return code; second, A710JOBB was released. After A710JOBB was finished, A710JOBd is released.

Example 6-17 Sample job group messages

```
11:55:56.16 JOB02578 00000080  ICH70001I LUTZ     LAST ACCESS AT 11:54:24 ON FRIDAY, JUNE 8, 2018
11:55:56.16 JOB02578 00000080  $HASP373 A710JOBA STARTED - WLM INIT  - SRVCLASS DFLT     - SYS SC80
11:55:56.17 JOB02578 00000080  Jobname  Procstep Stepname  CPU Time      EXCPs     RC
11:55:56.17 JOB02578 00000080  A710JOBA --None-- IEFBR1    00:00:00          8     00
11:55:56.17 JOB02578 00000080  $HASP395 A710JOBA ENDED - RC=0000
11:55:56.17 G0002577 00000080  $HASP1305 A710JOBC in job group LUTZ is flushed
11:55:56.17 JOB02579 00000080  $HASP1301 A710JOBB in job group LUTZ queued for execution
11:55:56.17 INTERNAL 00000280  SE '11.55.56 JOB02578 $HASP165 A710JOBA ENDED AT WTSCPLX8  MAXCC=0000'
                               ,LOGON,USER=(LUTZ)
11:55:56.17 INTERNAL 00000280  SE '11.55.56 JOB02581 $HASP165 A710JOBC ENDED AT WTSCPLX8 - FLUSHED',
                               LOGON,USER=(LUTZ)
11:55:56.17 JOB02579 00000080  ICH70001I LUTZ     LAST ACCESS AT 11:55:56 ON FRIDAY, JUNE 8, 2018
11:55:56.17 JOB02579 00000080  $HASP373 A710JOBB STARTED - WLM INIT  - SRVCLASS DFLT     - SYS SC80
11:55:56.18 JOB02579 00000080  Jobname  Procstep Stepname  CPU Time      EXCPs     RC
11:55:56.18 JOB02579 00000080  A710JOBB --None-- IEFBR1    00:00:00          8     00
11:55:56.18 JOB02579 00000080  $HASP395 A710JOBB ENDED - RC=0000
11:55:56.18 JOB02582 00000080  $HASP1301 A710JOBD in job group LUTZ queued for execution
11:55:56.18 INTERNAL 00000280  SE '11.55.56 JOB02579 $HASP165 A710JOBB ENDED AT WTSCPLX8  MAXCC=0000'
                               ,LOGON,USER=(LUTZ)
11:55:56.19 JOB02582 00000080  ICH70001I LUTZ     LAST ACCESS AT 11:55:56 ON FRIDAY, JUNE 8, 2018
11:55:56.19 JOB02582 00000080  $HASP373 A710JOBD STARTED - WLM INIT  - SRVCLASS DFLT     - SYS SC80
11:55:56.19 JOB02582 00000080  Jobname  Procstep Stepname  CPU Time      EXCPs     RC
11:55:56.19 JOB02582 00000080  A710JOBD --None-- IEFBR1    00:00:00          8     00
11:55:56.19 JOB02582 00000080  $HASP395 A710JOBD ENDED - RC=0000
```

```
11:55:56.19 G0002577 00000080  $HASP1304 job group LUTZ is complete
11:55:56.19 INTERNAL 00000280  SE '11.55.56 JOB02582 $HASP165 A710JOBD ENDED AT WTSCPLX8  MAXCC=0000'
                               ,LOGON,USER=(LUTZ)
```

After migrating all your jobs by using DJC, you should disable that function in your JES3 environment to prevent the future usage.

DEADLINE

To identify those job candidates for DEADLINE scheduling in your system, we suggest scanning your OPERLOG or DLOG for the last year for such messages, as shown in the following example:

```
IAT7401 DEADLINE DSP UNABLE TO COMPLETE FAILURE PROCESSING AFTER ABEND
IAT7405 INVALID COMMAND TO DEADLINE
IAT7410 DEADLINED JOBS ARE STILL IN THE SYSTEM.
IAT7415 JOB jobname (id) HAS INVALID DEADLINE TYPE(t), DEADLINE ENTRY NOT UPDATED.
IAT7420 START DEADLINE COMMAND ACCEPTED
IAT7425 JOB - IS PAST ITS DEADLINE
IAT7430 ALGORITHM - t - RUNNING FOR JOB jobname (id)
IAT7440 ERROR READING DEADLINE QUEUE, ALL ENTRIES LOST
IAT7445 ERROR READING DEADLINE QUEUE, UNDETERMINED NUMBER OF ENTRIES LOST
IAT7450 JOB jobname (id) PURGED
IAT7451 JOB jobname (id) IN PURGE WITH UNPROCESSED INTRDR JOBS, REPLY WAIT OR
CONTINUE
IAT7452 INCORRECT REPLY
IAT7455 OSE PURGE ERROR FOR JOB jobname (id)
```

If the investigation is complete, analyze the output and extract job names and the assigned user ID of that job. Now, you can address the need of transforming those jobs straight to the affected users by using the jobs user ID.

The professional BATCH scheduling system is sometimes not useful or flexible, especially for engineering jobs. Because many installations in the field use JES3 DEADLINE scheduling to release jobs in a timely manner, a replacement for DEADLINE under JES2 is needed.

JES2 provides a basic equivalent to the JES3 DEADLINE function. The JES2 SCHEDULE function can be used in several ways to release jobs for a certain time or date.

> **Important:** We offer you a free of charge REXX-based tool that runs as a server in a sysplex and replaces the JES3 DEADLINE Function. Your JCL DEADLINE user can continue to run with its JES3 DEADLINE card and run unchanged under JES2.

A JES2 JCL that uses the new SCHEDULE function is shown in Example 6-18.

Example 6-18   Example of JES2 JCL SCHEDULE

```
//A710JES2 JOB (RJO89350,LIPA,,1157),MSGCLASS=T,MSGLEVEL=(1,1),
//    CLASS=M,NOTIFY=&SYSUID
// SCHEDULE HOLDUNTL=('12:50',05/28/2018),STARTBY=('13:00',05/28/2018)
// SCHEDULE HOLDUNTL=('12:40',05/28/2018)
// SCHEDULE STARTBY='+01:00'
//*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
//IEFBR1   EXEC PGM=IEFBR14
//
```

Attention: The JCL that is shown in Example 6-18 demonstrates the power of the SCHEDULE statements, but you can use only one statement for one job.

You can code a certain date and time when a job is intended to run, or you can code a displacement time. If the assigned time is past the current day, it runs on the next day. Consider the following points:

- The first SCHEDULE holds the job until 05/28/2018 12:50. After passing that time stamp, it is released by JES2. JES2 attempts in parallel to run by the latest 13:00 at the same day by increasing its priority, if needed.
- The second SCHEDULE holds the job until 05/28/2018 12:40. After passing that time stamp, it is released by JES2. JES2 does not try any other action to promote the job. Based on the system utilization, it cannot be guaranteed that the job is running then.
- The third SCHEDULE releases the job 1 hour later than the job was submitted to the system.

Restriction: The new JES2 DEADLINE support replaces most of the JES3 DEADLINE functions. As of this writing, it is not possible to use the cycle function for jobs. Therefore, the WEEKLY, MONTHLY, and YEARLY options are not yet available. You can disable MAIN DEADLINE processing under JES2 by setting up JECLDEF JES3=(MAIN=IGNORE).

Main Device Scheduling

Because JES2 does not provide an equivalent function to Main Device Scheduling (MDS), you must prepare before any migration to JES2 to eliminate its use in JES3. If you do use MDS, it is likely that it is only used for tape.

If you use tape virtualization, it is reasonable to assume that you have more virtual tape drives than you use at one time; therefore, disabling MDS likely has no visible effect on job throughput. Even so, it is prudent to make this change before the migration so that if it does cause a problem, you can re-enable MDS while you investigate ways to address the problem.

Important: If you use MDS to control the order of running jobs or control job dependencies, JES2 does *not* provide an equivalent function. You must eliminate its use *before* you migrate to JES2.

Job throughput problems can be addressed by using the following methods:

- Remove “Tape and Disk setup” high water mark thresholds:
  - Use WLM and job schedulers to manage this job throughput instead.
  - Under the STANDARDS section in the JES3 initialization deck, replace SETUP=THWS with SETUP=NONE. This configuration helps make the JES3 more JES neutral.
- Convert from JES3-managed volumes to SMS-managed volumes.

Attention: All of the `*I S JES3` commands no longer work when SETUP=NONE is in place in your JES3 configuration. You should remove these commands from system automation and user-written REXX programs.

With this change, jobs are no longer checked in advance for all resources that they need. Therefore, jobs start, such as under JES2, and obtain data set allocation at the time they needed.

> **Information:** Checking for resources for jobs in advance is not a problem. You should consider only that you might see longer elapsed times for your jobs based on the availability of the resources that the jobs need. You might also see more ENQ contention because of jobs that are waiting for resources to become available.

Disk reader

The disk reader function submits JCL from a PDS to the internal reader by using a JES3 command. Many parameters are available to control the set of submitted jobs.

> **Information:** The disk reader facility (DR) is enabled by placing a //JES3DRDS DD card in the JES3 PROC or by specifying DYNALLOC,DDN=JES3DRDS,DSN=dsn in the JES3 inish deck. It is started by using the **`*X DR M=`** command, so a search in SYSLOG might be required to determine whether this facility is being used.

Starting with z/OS V2R4, you can use the new diskreader support that is provided by IBM. This support is similar to the JES3 diskreader support. To enable the support, add two statements in your JES2 initialization member (see Example 6-19).

Example 6-19 Diskreader enablement

```
SUBMITLIB(SLTEST) DD(01)=(DSNAME=LUTZ.SUBLIB.TEST),UNCONDITIONAL
SUBMITLIB(SLPROD) DD(01)=(DSNAME=LUTZ.SUBLIB.PROD),UNCONDITIONAL

SUBMITRDR AUTH=(DEVICE=NO,JOB=NO,SYSTEM=NO),
          CLASS=A,DD_DEFAULT=SLTEST,HOLD=NO,
          PRTYINC=01,PRTYLIM=08,

TRACE=NO
```

You add all regular z/OS data sets you want to be used for the JES2 diskreader. The SUBMITLIB statement follows the known PROCLIB initialization statement. Optionally, you also can code a UNIX System Services path that points to a directory that contains JCL.

> **Note:** You can use more than one SUBMITLIB statements in your JES2 initialization member to separate test and production data sets. They must have different DD names.

How the diskreader support is working is shown in Example 6-20.

Example 6-20 Submitting a job with diskreader

```
$SUBMIT,M=IEBGENER,HOLD=NO,DD=SLIBDD
$HASP000 OK
$HASP100 LUTZIEB  ON INTRDR      LUTZ                 FROM $SUBMIT
IEBGENER
IRR010I  USERID LUTZ     IS ASSIGNED TO THIS JOB.
ICH70001I LUTZ     LAST ACCESS AT 11:24:39 ON TUESDAY, JUNE 11, 2019
$HASP373 LUTZIEB  STARTED - INIT 1    - CLASS D        - SYS SC74
Jobname  Procstep Stepname  CPU Time      EXCPs     RC
LUTZIEB  --None-- COPY      00:00:00         44     00
$HASP395 LUTZIEB  ENDED - RC=0000
```

The restrictions for JES3 options that are listed in Table 6-8 apply to the JES2 $SUBMIT command and are not supported.

Table 6-8   JES3 options that are not supported

Parameter	Description
IN=	Device group for output
B=	Batch job size (in terms of job)
H/HN	Control-card processor hold (can hold the submitted jobs)
J=	Name of jobs in the member of where to start processing
JOBS=	Number of jobs to process from the member
K/KN	Keep reader after hitting EOF
P=	Priority of the control-card processor
PARMID=	Set of C/I options

The functions that are listed in Table 6-8 are rarely used in customer environments and should not affect a JES2 migration project.

JES3 DLOG

The JES3 DLOG function must be removed before you migrate to JES2. JES2 operates only with the z/OS OPERLOG function. Therefore, start the migration from DLOG to OPERLOG as soon as possible.

An example DLOG that starts a JES3 printer is shown in Example 6-21.

Example 6-21   JES3 DLOG example

```
MLG          12340 1329067  S22 R= MP20NVC  NVS22    Z000507  ! NVC-CMDIF: Z000507 S22->S22 *S A100 WC=1
     A0050722 12340 1329067 +S A100 WC=1
     A0050722 12340 1329067  IAT7089 WTR      (JOB14567) ON A100     (    )
     A0050722 12340 1329067  IAT7075 STARTED
MLG          12340 1329067 &IAT7001 JOB PVS8259P (JOB57960) IS ON WRITER A100(    ),RECORDS=86,PAGES=18
```

The equivalent of starting a JES2 printer by using the OPERLOG function is shown in Example 6-22.

Example 6-22   OPERLOG example

```
NC0000000 S28      18166 15:58:35.72 Z000426  00000210  $SPRT131
NR0000000 S28      18166 15:58:35.73 Z000426  00000010  $HASP000 OK
NC0000000 S28      18166 15:58:35.73 INTERNAL 00000210  START  PSFGRP6I.PSFGRP6I,,,( ),SUB=JES2
```

The main differences that you see is the message header of both examples. Different information is available and in another location inside the message header. For example, the time is in another location, uses another format, and includes more information in OPERLOG than in DLOG.

Important: Before you can migrate DLOG to OPERLOG, you must analyze which tools or custom-made programs that use DLOG must be converted to OPERLOG capabilities.

First, enable OPERLOG in your system according to the HARDCOPY option in your CONSOLxx member. Then, you can disable JES3 OPERLOG by using the following command:

```
*F O DLOG=OFF
```

6.7 Transforming JCL and JECL

Transforming the JCL and JECL in your installation is a large task. Although it is not a complex process, many stakeholders are involved because of the sensitive nature of the process.

> **Information:** The intention of migrating our jobs was to align the JCL and JECL in a way that works with both JES versions.

We separated the transformation task into the following subtasks:

- For the production JCL that is controlled by a professional BATCH scheduling system, we aligned JES3 JCL to a common form that is usable with both JES versions in front of the migration.
- We provided support for users to convert their own private libraries to a JES2-conforming version.

Before you begin, inspect your production JCL data sets for the occurrence of JES3 JECL cards. For more information about all possible JES3 JECLs, see Chapter 4, “JES2 functions to help migration” on page 43.

The next step can find a replacement of that JES3 JECL. In the most current release of z/OS, most of the JES3 JECL statements are honored and processed, if needed. Therefore, JES3 JECLs do not need to be converted to their JES2 equivalent (if they exist).

The support levels for JES3 JECLs are listed in Table 6-9.

Table 6-9 Possible JES2 support options

Support	Description
Obsolete	A warning diagnostic is written, but is otherwise ignored.
Not supported	An error message is generated and the job is given a JCL ERROR.
Supported	Full support is provided under JES2.

The most current support of JES3 JECL cards in your JCL is listed in Table 6-10.

Table 6-10 JES3 JECL support

JES3 JECL	Sub option	Support	Comment
//*DATASET		Not supported	Is tolerated but ignored
//*ENDDATASET		Not supported	Required if //*DATASET is present
//*ENDPROCESS		Not supported	

JES3 JECL	Sub option	Support	Comment
//*FORMAT	DDNAME CARRIAGE/FCB CHARS COMPACT COPIES DEST EXTWTR FLASH FORMS MODIFY PRTY STACKER TRAIN	Supported	Each //*FORMAT statement results in a // OUTPUT statement, which is forced to be after the JOB statement and before the first EXEC statement. The name that is given the OUTPUT statements uses the form JES2*nnnn*, where *nnnn* begins at 0000.
	CHNSIZE INT OVFL THRESHLD	Not supported	
//*MAIN	ACMAIN IORATE LREGION MSS RINGCHK TRKGRPS TYPE	Obsolete	
	BYTES CARDS CLASS HOLD JOURNAL LINES ORG PAGES PROC SYSTEM USER	Supported	
	DEADLINE EXPDTCHK FAILURE FETCH SETUP SPART THWSSEP UPDATE	Not supported	DEADLINE is replaced by JES2 SCHEDULE statement.
//*NET	ID/NETID ABCMP/AC ABNORMAL NORMAL NETREL/NR NHOLD/HC NRCMP/PC OPHOLD/OH RELEASE/RL	Supported	Converted internally to a JES2 JOBGROUP.

JES3 JECL	Sub option	Support	Comment
	DEVPOOL, DEVRELSE RELSCHCT/RS	Obsolete	
//*NETACCT		Supported	All keywords that are supported in the same way as JES3 supports them.
//*OPERATOR		Supported	Message text ends in 71, not 80.
//*PAUSE		Not supported	Is ignored if present.
//*PROCESS		Not supported	
//*ROUTE	XEQ	Supported	No support for an alternative JES2 /*ROUTE XEQ.
//*SIGNON		Not supported	
//*SIGNOFF		Not supported	

If you use JES3 JECL statements in your JCL that do not include an equivalent under JES2, these statements must be changed manually. To give users of such JES3 functions the ability to convert their JCL to a JES2 conforming JCL, we provide a sample REXX executable that addresses this need (for more information, see Appendix A, “Sample JES3 exit to analyze JECL usage” on page 195). This sample code demonstrates how to convert such functions in JCL and must be adjusted based on your needs.

Attention: At this point, unusual converted JCL and JECL were observed. Therefore, it is recommended that your system is closely monitored after migration. For more information, see 6.12, “Hints and tips” on page 172.

The REXX program transforms the JECL in the same way as the professional program that we used for the production JCL.

What happens if jobs are not changed?

All JES3 JECL cards that begin with //* are processed by JES2. Therefore, these jobs will not fail at any time, which results in JES3 JECL (which is not yet supported by IBM) being ignored and treated as a comment.

Your JES2 settings determine the behavior of your JCL and how JES3 JECL cards are handled. By default (or if you do not code anything), all JES3 JECL cards are processed by JES2.

For most JES3 JECL statements, you can control how JES2 handles that statement.

Keywords exist for each JECL card type. Each keyword includes the valid options that are listed in Table 6-11.

Table 6-11 JES3 JECL process options

Option	Description
PROCESS	The specific JES3 JECL statement is processed (translated or directly processed). This option is the default.
WARN	The specific JES3 JECL statement is processed, but a warning message is issued that indicates that the installation intends to discontinue use of this statement in the future and that it should no longer be used.
FAIL	An error message is generated for the specific JES3 JECL statement. The job does not run.
IGNORE	The specific JES3 JECL statement is ignored and treated as a comment.

6.8 Migrating system automation

Strong monitoring is available for JES3. This monitoring addresses the need to quickly detect dangerous situations during the JES's lifetime.

First, create a list of all monitored items that are active in your system automation, which might include the following items:

- Alarms that set for certain JES3 messages.
- Custom REXX programs that perform any kind of JES3 monitoring.
- Native JES3 commands that are issued frequently.
- Custom REXX programs to conduct predefined tasks; for example, moving JES3 GLOBAL to another system.

Upon completion, assign the following tasks that must be done for migration:

- Removal of the function; for example, all items that are related to JES3 specialities (JES3 GLOBAL).
- Transfer the function to a JES2 version.
- Find the appropriate JES2 message for JES3 to monitor and add any JES2 messages that you want to be monitored.

6.8.1 New JES2 messages

To achieve the same stability that you likely had under JES3, you might need to add JES2 messages to your system automation. The messages that are listed in Table 6-12 on page 154 are intended to ensure that JES2 is starting without any operator response. These messages are only a recommendation and can vary in your installation.

Attention: This process is ongoing and must be reviewed frequently. It has no claim to completeness.

Table 6-12 JES2 automated startup

Message	Description	Reply
HASP405	JES2 IS UNABLE TO DETERMINE IF OTHER MEMBERS ARE ACTIVE	Y
HASP417	ARE THE VALUES JES2 WILL USE CORRECT?	Y
HASP420	REPLY ‘ Y’ IF memname IS ALL MEMBERS ARE DOWN (IPL REQUIRED), ‘N’IF NOT	Y
HASP426	SPECIFY OPTIONS – JES2 jeslevel SSNAME= ssname	Y
HASP434	INVALID CHECKPOINT RECORD ON CKPTn DATA SET This message ends up in the JES2 Checkpoint reconfiguration dialog.	Y
HASP454	SHOULD JES2 BYPASS THE MULTI-MEMBER INTEGRITY LOCK? (‘Y’ OR ‘ N’)	Y

For the beginning when JES2 is used, you can put the messages that are listed in Table 6-13 in your system automation. This inclusion should prevent the most important JES2 cases that influence the stability of JES2.

Table 6-13 JES2 messages to monitor

Message	Description
$HASP050	JES2 RESOURCE SHORTAGE OF resource-type nnn %UTILIZATION REACHED
$HASP065	AWAITING RESPONSE TO HASPxxx MESSAGE, AUTO-REPLY N sec SECONDS
$HASP080	JES2 SYSTEM DUMP REQUESTED FROM mod (adr) + X'oooooo'
$HASP094	I/O ERROR ON SPOOL, MTTR=nnnnnnnn
$HASP095	JES2 CATASTROPHIC ERROR CODE=cde (RC= rsnc) JES2 CATASTROPHIC ABEND CODE=cde (RC= rsnc)
$HASP110	jobname illegal JOB card reason jobname Invalid JOB statement reason jobname Illegal //*MAIN card reason
$HASP121	jobname device name ERROR RECEIVING NETWORK JOB HEADER RC=rc jobname device name ERROR RECEIVINGNETWORK DATA SET HEADER RC=rc jobname device name ERROR RECEIVINGNETWORK JOB TRAILER RC=rc
$HASP198	REPLY TO HASP098 WITH ONE OF THE FOLLOWING: Message that appears during abnormal end of JES2
$HASP263	WAITING FOR ACCESS TO JES2 CHECKPOINT VOLUME volser LOCK HELD BY MEMBER member_name WAITING FOR ACCESS TO JES2 CHECKPOINT VOLUME volserLOCK HELD BY SYSTEM WAITING FOR ACCESS TO JES2 CHECKPOINT VOLUME volserSYSTEM MANAGED PROCESS ACTIVE
$HASP292	MEMBER member-name JES2 WAITING FOR RESPONSE TO READ FROM ckpt MEMBER member-name JES2 WAITING FOR RESPONSE TO WRITE TO ckpt MEMBER member-name JES2 WAITING FOR RESPONSE TO FORMAT OF ckpt MEMBER member-name JES2 WAITING FOR RESPONSE TO EXTEND OF ckpt MEMBER member-name JES2 WAITING FOR RESPONSE TO I/O TO ckpt
$HASP310	Job name TERMINATED AT END OF MEMORY
$HASP355	SPOOL VOLUMES ARE FULL

Message	Description
$HASP375	Job name ESTIMATED metrics EXCEEDED job name ESTIMATE EXCEEDED BY nnn metrics xxx% SPOOL
$HASP490	HOT START DENIED - RE-IPL REQUIRED
$HASP492	The general JES2 start message that indicates JES2 is started.
$HASP496	This message indicates a mismatch between the saved initialization in the checkpoint compared with the JES2 initialization PARMLIB configuration during JES2 startup.
$HASP500	CONNECTION CONTROL RECEIVE ON lna text
$HASP531	Job name: Devname INVALID DATA BLOCK DETECTED TRANSMITTER devname ON NODE nodename
$HASP543	Jobname: Devname DELETED
$HASP565	General message that indicates that no NJE connection was established to the partner node
$HASP9156	ADDRESS SPACES WAITING FOR SPOOL SPACE
$HASP9162	PCES WAITING FOR SPOOL SPACE
$HASP9201	JES2 MAIN TASK WAIT DETECTED AT module+offset DURATION-hh:mm:ss.xx PCE pcename EXIT exit JOB ID jobid COMMAND jes2_command
$HASP9207	This message can indicate that an alert or an incident JES2 is tracking.

6.8.2 CKPT reconfiguration

In JES2 MAS, all participating systems need access to the checkpoint. This checkpoint is a sensitive resource and it is available twice for security reasons. Each checkpoint includes a backup that is defined to JES2, as shown in Example 6-23.

Example 6-23 Sample checkpoint configuration

```
$DCKPTDEF
$HASP829 CKPTDEF
$HASP829 CKPTDEF  CKPT1=(STRNAME=JES2CKPT_1,INUSE=YES,VOLATILE=YES),
$HASP829          CKPT2=(DSNAME=JES2#A.RZ4.PO.CKPT2,VOLSER=SYA410,
$HASP829          INUSE=YES,VOLATILE=NO),

$HASP829          NEWCKPT1=(DSNAME=JES2#A.RZ4.PO.CKPT1NEW,
$HASP829          VOLSER=SYA412)
$HASP829          NEWCKPT2=(DSNAME=JES2#A.RZ4.PO.CKPT2NEW,
$HASP829          VOLSER=SYA411),MODE=DUPLEX,DUPLEX=ON,LOGSIZE=1,
$HASP829          VERSIONS=(STATUS=ACTIVE,NUMBER=50,WARN=80,MAXFAIL=0,
$HASP829          NUMFAIL=0,VERSFREE=50,MAXUSED=2),RECONFIG=NO,
$HASP829          VOLATILE=(ONECKPT=IGNORE,ALLCKPT=WTOR),OPVERIFY=NO
```

If an issue exists in accessing one of the primary checkpoints, the first or second JES2 automatically forwards this checkpoint to the defined backup checkpoint, as listed in Table 6-14.

Table 6-14 Sample checkpoint definitions

Checkpoint	Primary	Backup
One	`JES2CKPT_1`	`JES2#A.RZ4.PO.CKPT1NEW`
Two	`JES2#A.RZ4.PO.CKPT2`	`JES2#A.RZ4.PO.CKPT2NEW`

If such a situation occurs, you run without any backup checkpoint available. To avoid this situation, monitor the JES2 message as listed in Table 6-15.

Table 6-15 Checkpoint Actions

Message	Checkpoint	Action
HASP280	CF backup is active	Add the previous primary as backup to the system $TCKPTDEF,NEWCKPT1=(STRNAME=JES2CKPT_1)
HASP280	CF primary is active	Add the backup checkpoint to the system $TCKPTDEF,NEWCKPT1=(STRNAME=JES2CKPT_1_NEW)

6.8.3 Replacement for JES3 unique functions

One of the differences between JES3 and JES2 is the ability to define more options to sysout classes, such as DESTINATION, as shown in Example 6-24.

Example 6-24 Sample JES3 output class definitions

```
SYSOUT,CLASS=0,OVFL=OFF,HOLD=EXTWTR,DEST=LOCAL
SYSOUT,CLASS=P,OVFL=OFF,DEST=RZ2
SYSOUT,CLASS=Q,DEST=RZ2
```

Under JES2, such options for output classes are not available. In the customer project, we implemented a solution that is based on system automation.

A timer periodically sends a JES2 command to the system that changes the destination of certain output elements. Some JES2 sample commands from the customer environment are listed in Table 6-16. These commands change all output elements in the specified sysout class.

Table 6-16 Sample JES2 commands

JES2 command	Description
$TO JOBQ,/Q=0,/D=LOCAL,D=<Target>	Transfers all output elements in sysout class 0 and destination local to the Target sysplex.
$TO JOBQ,/Q=T,/AGE>3,Q=O	All output elements in sysout class T are moved to class 0 if older than three days.

Attention: If you must transfer many output elements, consider issuing the commands more often. Doing so avoids the situation in which JES2 is busy for extended periods in processing the request.

By using this solution, you can transfer entire job classes to another system. If you must transfer output elements to another system but must keep the old destination, you cannot use JES2 system commands. The use of these commands results in loosing such information. An example in which a simple REXX program is used is shown in Example 6-25.

Example 6-25 REXX for transferring output

```
/*REXX================================================================*/
/* Purpose: Transfer all Output elements located in Sysout class 2    */
/*          to sysplex PLEXQ and keep to old destination (the old     */
/*          printer name).                                            */
/*                                                                    */
/* History: 14.06.18 LK Initial for ITSO Redbooks                     */
/*                                                                    */
/*====================================================================*/
 rc = ISFCalls('ON')

 ISFPrefix = "**"
 ISFOwner  = "**"
 DestPlex  = 'PLEXQ'
 Address SDSF "ISFEXEC O"
 Do i=1 to JNAME.0
  if SCLASS.i = '2'
   Then Do
     Address SDSF "ISFACT O TOKEN('"TOKEN.i"')",
              "PARM(DEST" DestPlex!!"."!!DEST.i")"
    End
  End

  rc = ISFCalls('OFF')
 exit
```

The differences while transferring output to another system by using the JES2 command and the REXX program are listed in Table 6-17.

Table 6-17 DEST comparison

Output element at origin	In PLEXQ with $TOJOBQ command	In PLEXQ with the REXX program
DEST=B433	DEST=LOCAL	DEST=B433

Change this REXX program to suit your needs and place it in your system automation based on your needs. The program should run with more or less frequency.

6.9 Migrating security

Complete the following steps to migrate your security definitions to a JES2 version:

1. Convert existing JES3 prefixed profiles in RACF to a JES2 prefixed profile. The following RACF classes are affected:
   - Class NODES and WRITER for NJE and RJE definitions
   - JESINPUT for offloading
   - JESSPOOL for controlling job permissions

2. Add RACF profiles for all JES2 commands.
3. Add RACF profiles for all SDSF/EJES JES2 profiles.
4. Add new RACF profiles for any printer you might use.

6.9.1 JES3 prefixed profiles

For all JES3-related RACF profiles, define a JES2 prefixed equivalent in your RACF database. Almost all RACF profiles that are used for JES3 are the same under JES2. For more information about exceptions, see E.1.1, “RACF profiles used by exits” on page 235.

6.9.2 New JES2 command profiles

The main part of migrating to JES2 are the profiles for all JES2 commands. All of these profiles should have their appropriate RACF profile defined.

It is recommended to place all of the profiles that are listed in Table 6-18 in your RACF database to ensure that all JES2 system commands are protected. You replace only the `.jesx` prefix with your own JES2 subsystem ID (usually JES2).

Table 6-18 Profiles in RACF database

Command	Command option	RACF profile	Access level
$A		jesx.MODIFYREALEASE.*	UPDATE
	$A A	jesx.MODIFYRELEASE.JOB	UPDATE
	$A J	jesx.MODIFYRELEASE.BAT	UPDATE
	$A JOBQ	jesx.MODIFYRELEASE.JST	UPDATE
	$A S	jesx.MODIFYRELEASE.STC	UPDATE
	$A T	jesx.MODIFYRELEASE.TSU	UPDATE
$ACTIVATE		jesx.ACTIVATE.FUNCTION	CONTROL
	$ACTIVATE	jesx.ACTIVATE.FUNCTION	CONTROL
$ADD		jesx.ADD.*	CONTROL
	$ADD APPL	jesx.ADD.APPL	CONTROL
	$ADD CONNECT	jesx.ADD.CONNECT	CONTROL
	$ADD DESTID	jesx.ADD.DESTID	CONTROL
	$ADD FSS	jesx.ADD.FSS	CONTROL
	$ADD LINE	jesx.ADD.LINE	CONTROL
	$ADD LOADMOD	jesx.ADD.LOADMOD	CONTROL
	$ADD LOGON	jesx.ADD.LOGON	CONTROL
	$ADD NETSRV	jesx.ADD.NETSRV	CONTROL
	$ADD PROCLIB	jesx.ADD.PROCLIB	CONTROL
	$ADD PRTnnnn	jesx.ADD.DEV	UPDATE
	$ADD REDIRECT	jesx.ADD.REDIRECT	CONTROL

Command	Command option	RACF profile	Access level
	$ADD RMT	jesx.ADD.RMT	CONTROL
	$ADD SOCKET	jesx.ADD.SOCKET	CONTROL
	$ADD SUBMITLIB	jesx.ADD.SUBMITLIB	CONTROL
	$ADD SRVCLASS	jesx.ADD.SRVCLASS	CONTROL
$B	device	jesx.BACKSP.DEV	UPDATE
$C A**		jesx.CANCEL.AUTOCMD	CONTROL
$C J/S/T O		jesx.CANCEL.JST*/BAT*/STC*/TSU*	
	$C J	jesx.CANCEL.BAT	UPDATE
	$C S	jesx.CANCEL.STC	UPDATE
	$C T	jesx.CANCEL.TSU	UPDATE
	$C O J	jesx.CANCEL.BATOUT	UPDATE
	$C O JOBQ	jesx.CANCEL.JSTOUT	UPDATE
	$C O S	jesx.CANCEL.STCOUT	UPDATE
	$C O T	jesx.CANCEL.TSUOUT	UPDATE
$C device		jesx.CANCEL.DEV	UPDATE
	$C Lx.yy	jesx.CANCEL.DEV	UPDATE
	$C device	jesx.CANCEL.DEV	UPDATE
	$C OFFn.JR	jesx.CANCEL.DEV	UPDATE
	$C OFFn.JT	jesx.CANCEL.DEV	UPDATE
	$C OFFn.SR	jesx.CANCEL.DEV	UPDATE
	$C OFFn.ST	jesx.CANCEL.DEV	UPDATE
$D *		jesx.DISPLAY.*	READ
	$D A	jesx.DISPLAY.JOB	READ
	$D ACTIVATE	jesx.DISPLAY.ACTIVATE	READ
	$D ACTRMT	jesx.DISPLAY.ACTRMT	READ
	$D APPL	jesx.DISPLAY.APPL	READ
	$D CKPTDEF	jesx.DISPLAY CKPTDEF	READ
	$D CONDEF	jesx.DISPLAY.CONDEF	READ
	$D CONNECT	jesx.DISPLAY.CONNECT	READ
	$D DESTDEF	jesx.DISPLAY.DESTDEF	READ
	$D DEStid	jesx.DISPLAY.DESTID	READ
	$D F	jesx.DISPLAY.QUE	READ
	$D I	jesx.DISPLAY.INITIATOR	READ
	$D INITINFO	jesx.DISPLAY.INITINFO	READ

Command	Command option	RACF profile	Access level
	$D JES2	jesx.DISPLAY.SYS	READ
	$D J	jesx.DISPLAY.BAT	READ
	$D JOBQ	jesx.DISPLAY.JST	READ
	$D JOBCLASS	jesx.DISPLAY.JOBCLASS	READ
	$D L(nnnn).JR(n)	jesx.DISPLAY.L	READ
	$D L(nnnn).JT(n)	jesx.DISPLAY.L	READ
	$D L(nnnn).SR(n)	jesx.DISPLAY.L	READ
	$D L(nnnn).ST(n)	jesx.DISPLAY.L	READ
	$D LINE	jesx.DISPLAY.LINE	READ
	$D LOADmod	jesx.DISPLAY.LOADMOD	READ
	$D MASDEF	jesx.DISPLAY.MASDEF	READ
	$D MEMBer	jesx.DISPLAY.SYS	READ
	$D MODULE	jesx.DISPLAY.MODULE	READ
	$D N	jesx.DISPLAY.JOB	READ
	$D NETSRV	jesx.DISPLAY.NETSRV	READ
	$D NJEDEF	jesx.DISPLAY.NJEDEF	READ
	$D NODE	jesx.DISPLAY.NODE	READ
	$D O J	jesx.DISPLAY.BATOUT	READ
	$D O JOBQ	jesx.DISPLAY.JSTOUT	READ
	$D O S	jesx.DISPLAY.STCOUT	READ
	$D O T	jesx.DISPLAY.TSUOUT	READ
	$D OPTSDEF	jesx.DISPLAY.OPTSDEF	READ
	$D PATH	jesx.DISPLAY.PATH	READ
	$D PCE	jesx.DISPLAY.PCE	READ
	$D PRT	jesx.DISPLAY.DEV	READ
	$D PRTnnnn	jesx.DISPLAY.DEV	READ
	$D PUNnn	jesx.DISPLAY.DEV	READ
	$D Q	jesx.DISPLAY.JOB	READ
	$D REBLD	jesx.DISPLAY.REBLD	READ
	$D RDI	jesx.DISPLAY.RDI	READ
	$D RDRnn	jesx.DISPLAY.DEV	READ
	$D Rnnnnn.CON	jesx.DISPLAY.DEV	READ
	$D Rnnnnn.PRm	jesx.DISPLAY.DEV	READ
	$D Rnnnnn.PUm	jesx.DISPLAY.DEV	READ

Command	Command option	RACF profile	Access level
	$D Rnnnnn.RDm	jesx.DISPLAY.DEV	READ
	$D REDIRect	jesx.DISPLAY.REDIRECT	READ
	$D S	jesx.DISPLAY.STC	READ
	$D SOCKET	jesx.DISPLAY.SOCKET	READ
	$D SPOOL	jesx.DISPLAY.SPOOL	READ
	$D SPOOLDEF	jesx.DISPLAY.SPOOLDEF	READ
	$D SUBMITLIB	jesx.DISPLAY.SUBMITLIB	READ
	$D SUBMITRDR	jesx.DISPLAY.SUBMITRDR	READ
	$D SRVCLASS	jesx.DISPLAY.SRVCLASS	READ
	$D SSI	jesx.DISPLAY.SSI	READ
	$D SUBNET	jesx.DISPLAY.SUBNET	READ
	$D T	jesx.DISPLAY.TSU	READ
	$D TRACE(x)	jesx.DISPLAY.TRACE	READ
	$D U	jesx.DISPLAY.DEV	READ
	$D init stmt	jesx.DISPLAY.initstmt	READ
	$L J	jesx.DISPLAY.BATOUT	READ
	$L JOBQ	jesx.DISPLAY.JSTOUT	READ
	$L S	jesx.DISPLAY.STCOUT	READ
	$L T	jesx.DISPLAY.TSUOUT	READ
$D M		jesx.SEND.MESSAGE	READ
	$D M	jesx.SEND.MESSAGE	READ
$DEL		jesx.DEL.*	CONTROL
	$DEL CONNECT	jesx.DEL.CONNECT	CONTROL
	$DEL DESTID	jesx.DEL.DESTID	CONTROL
	$DEL LOADMOD	jesx.DEL.LOADMOD	CONTROL
	$DEL PROCLIB	jesx.DEL.PROCLIB	CONTROL
	$DEL SUBMITLIB	jesx.DEL.SUBMITLIB	CONTROL
$E CKPTLOCK	$E CKPTLOCK	jesx.RESTART.SYS	CONTROL
$E J	$E J	jesx.RESTART.BAT	CONTROL
$E JOBQ		jesx.RESTART.JST	CONTROL
	$E JOBQ	jesx.RESTART.JST	CONTROL
	$E LINE(x)	jesx.RESTART.LINE	CONTROL
	$E LOGON(x)	jesx.RESTART.LOGON	CONTROL
	$E MEMBER()	jesx.RESTART.SYS	CONTROL

Command	Command option	RACF profile	Access level
	$E NETSRV	jesx.RESTART.NETSRV	CONTROL
	$E OFFn.JT	jesx.RESTART.DEV	UPDATE
	$E OFFn.ST	jesx.RESTART.DEV	UPDATE
	$E device	jesx.RESTART.DEV	UPDATE
$F device		jesx.FORWARD.DEV	UPDATE
	$F device	jesx.FORWARD.DEV	UPDATE
$G *		jesx.G*	UPDATE
	$G A	jesx.GMODIFYRELEASE.JOB	UPDATE
	$G C	jesx.GCANCEL.JOB	UPDATE
	$G D	jesx.GDISPLAY.JOB	READ
	$G H	jesx.GMODIFYHOLD.JOB	UPDATE
	$G R	jesx.GROUTE.JOBOUT	UPDATE
	$G R (for execution)	jesx.GROUTE.JOBOUT	UPDATE
$H		jesx.MODIFYHOLD.*	UPDATE
	$H A	jesx.MODIFYHOLD.JOB	UPDATE
	$H J	jesx.MODIFYHOLD.BAT	UPDATE
	$H JOBQ	jesx.MODIFYHOLD.JST	UPDATE
	$H S	jesx.MODIFYHOLD.STC	UPDATE
	$H T	jesx.MODIFYHOLD.TSU	UPDATE
$I device		jesx.INTERRUPT.DEV	UPDATE
	$I device	jesx.INTERRUPT.DEV	UPDATE
$J*		jesxMON.*	CONTROL
	$JD DETAILS	jesxMON.DISPLAY.DETAIL	READ
	$JD HISTORY	jesxMON.DISPLAY.HISTORY	READ
	$JD JES	jesxMON.DISPLAY.JES	READ
	$JD MONITOR	jesxMON.DISPLAY.MONITOR	READ
	$JD STATUS	jesxMON.DISPLAY.STATUS	READ
	$J STOP	jesxMON.STOP.MONITOR	CONTROL
$M		jesx.MSEND.CMD	READ
	$M	jesx.MSEND.CMD	READ
$M SPL		jesx.MIGRATE.FUNCTION	CONTROL
	$M SPL	jesx.MIGRATE.FUNCTION	CONTROL
$N		jesx.NSEND.CMD	READ
	$N	jesx.NSEND.CMD	READ

Command	Command option	RACF profile	Access level
$N device		jesx.REPEAT.DEV	UPDATE
	$N device	jesx.REPEAT.DEV	UPDATE
$O *		jesx.RELEASE.BATOUT	UPDATE
	$O J	jesx.RELEASE.BATOUT	UPDATE
	$O JOBQ	jesx.RELEASE.JSTOUT	UPDATE
	$O S	jesx.RELEASE.STCOUT	UPDATE
	$O T	jesx.RELEASE.TSUOUT	UPDATE
$P *		jesx.STOP.* but not jesx.STOP.JST*/BAT*/STC*/TSU*	CONTROL
	$P	jesx.STOP.SYS	CONTROL
	$P I	jesx.STOP.INITIATOR	CONTROL
	$P JES2	jesx.STOP.SYS	CONTROL
	$P LINE(x)	jesx.STOP.LINE	CONTROL
	$P LOGON(x)	jesx.STOP.LOGON	CONTROL
	$P NETSRV	jesx.STOP.NETSRV	CONTROL
	$P OFFn.JR	jesx.STOP.DEV	UPDATE
	$P OFFn.JT	jesx.STOP.DEV	UPDATE
	$P OFFn.SR	jesx.STOP.DEV	UPDATE
	$P OFFn.ST	jesx.STOP.DEV	UPDATE
	$P OFFLOADn	jesx.STOP.DEV	UPDATE
	$P RMT(x)	jesx.STOP.RMT	CONTROL
	$P SPOOL	jesx.STOP.SPOOL	CONTROL
	$P SRVCLASS	jesx.STOP.SRVCLASS	CONTROL
	$P TRACE(x)	jesx.STOP.TRACE	CONTROL
	$P XEQ	jesx.STOP.SYS	CONTROL
	$P device	jesx.STOP.DEV	UPDATE
$P J/B/T + O		jesx.STOP.JST*/BAT*/STC*/TSU*	
	$P JOBQ	jesx.STOP.JST	UPDATE
	$P JOB	jesx.STOP.BAT	UPDATE
	$P STC	jesx.STOP.STC	UPDATE
	$P TSU	jesx.STOP.TSU	UPDATE
	$PO JOBQ	jesx.STOP.JSTOUT	UPDATE
	$PO JOB	jesx.STOP.BATOUT	UPDATE
	$PO STC	jesx.STOP.STCOUT	UPDATE

Command	Command option	RACF profile	Access level
	$PO TSU	jesx.STOP.TSUOUT	UPDATE
$POLICY	$POLICY IMPORT	jesx.POLICY.DELETE	UPDATE
	$POLICY DISABLE	jesx.POLICY.DISABLE	UPDATE
	$POLICY DISPLAY	jesx.POLICY.DISPLAY	UPDATE
	$POLICY ENABLE	jesx.POLICY.ENABLE	UPDATE
	$POLICY IMPORT	jesx.POLICY.IMPORT	UPDATE
$R *		jesx.ROUTE.JOBOUT	UPDATE
	$R ALL	jesx.ROUTE.JOBOUT	UPDATE
	$R PRT	jesx.ROUTE.JOBOUT	UPDATE
	$R PUN	jesx.ROUTE.JOBOUT	UPDATE
	$R XEQ	jesx.ROUTE.JOBOUT	UPDATE
$S *		jesx.START.* but not jesx.START.BAT	CONTROL
	$S	jesx.START.SYS	CONTROL
	$S A	jesx.START.AUTOCMD	CONTROL
	$S I	jesx.START.INITIATOR	CONTROL
	$S LINE(x)	jesx.START.LINE	CONTROL
	$S LOGON(x)	jesx.START.LOGON	CONTROL
	$S N	jesx.START.NET	CONTROL
	$S NETSRV(nnn)	jesx.MODIFY.NETSRV	CONTROL
	$S OFFn.JR	jesx.START.DEV	UPDATE
	$S OFFn.JT	jesx.START.DEV	UPDATE
	$S OFFn.SR	jesx.START.DEV	UPDATE
	$S OFFn.ST	jesx.START.DEV	UPDATE
	$S OFFLOADn	jesx.START.DEV	UPDATE
	$S device	jesx.START.DEV	UPDATE
	$S RMT(x)	jesx.START.RMT	CONTROL
	$S SPOOL	jesx.START.SPOOL	CONTROL
	$S SRVCLASS	jesx.START.SRVCLASS	CONTROL
	$S TRACE(x)	jesx.START.TRACE	CONTROL
	$S XEQ	jesx.START.SYS	CONTROL
$S J	$S J	jesx.START.BAT	UPDATE
	$SUBMIT	jesx.SUBMIT.JOB	CONTROL
$T *		jesx.MODIFY.* but not jesx.MODIFY.JST*/BAT*/STC*/TSU*	CONTROL

Command	Command option	RACF profile	Access level
	$T A(CREATE)	jesx.MODIFY.AUTOCMD	READ
	$T A(OWNER)	jesx.MODIFY.AUTOCMD	READ
	$T A(NOT OWNER)	jesx.MODIFY.AUTOCMD	CONTROL
	$T APPL	jesx.MODIFY.APPL	CONTROL
	$T BUFDEF	jesx.MODIFY.BUFDEF	CONTROL
	$T CKPTDEF	jesx.MODIFY.CKPTDEF	CONTROL
	$T CONDEF	jesx.MODIFY.CONDEF	CONTROL
	$T CONNECT	jesx.MODIFY.CONNECT	CONTROL
	$T DEBUG	jesx.MODIFY.DEBUG	CONTROL
	$T DESTDEF	jesx.MODIFY.DESTDEF	CONTROL
	$T DEStid	jesx.MODIFY.DESTID	CONTROL
	$T ESTBYTE	jesx.MODIFY.ESTBYTE	CONTROL
	$T ESTIME	jesx.MODIFY.ESTIME	CONTROL
	$T ESTLNCT	jesx.MODIFY.ESTLNCT	CONTROL
	$T ESTPAGE	jesx.MODIFY.ESTPAGE	CONTROL
	$T ESTPUN	jesx.MODIFY.ESTPUN	CONTROL
	$T EXIT	jesx.MODIFY.EXIT	CONTROL
	$T FSS	jesx.MODIFY.FSS	CONTROL
	$T I	jesx.MODIFY.INITIATOR	CONTROL
	$T INTRDR	jesx.MODIFY.INTRDR	CONTROL
	$T JOBCLASS	jesx.MODIFY.JOBCLASS	CONTROL
	$T JOBDEF	jesx.MODIFY.JOBDEF	CONTROL
	$T JOBPRTY	jesx.MODIFY.JOBPRTY	CONTROL
	$T LINE	jesx.MODIFY.LINE	CONTROL
	$T LOADMOD	jesx.MODIFY.LOADMOD	CONTROL
	$T LOGON	jesx.MODIFY.LOGON	CONTROL
	$T MASDEF	jesx.MODIFY.MASDEF	CONTROL
	$T MEMBER(x)	jesx.MODIFY.SYS	CONTROL
	$T NETSRV	jesx.MODIFY.NETSRV	CONTROL
	$T NJEDEF	jesx.MODIFY.NJEDEF	CONTROL
	$T NODE	jesx.MODIFY.NODE	CONTROL
	$T NUM	jesx.MODIFY.NUM	CONTROL
	$T OFFLOADx	jesx.MODIFY.OFFLOAD	CONTROL
	$T OUTCLASS	jesx.MODIFY.OUTCLASS	CONTROL

Command	Command option	RACF profile	Access level
	$T OUTDEF	jesx.MODIFY.OUTDEF	CONTROL
	$T OUTPRTY	jesx.MODIFY.OUTPRTY	CONTROL
	$T PCE	jesx.MODIFY.PCE	CONTROL
	$T PRINTDEF	jesx.MODIFY.PRINTDEF	CONTROL
	$T device	jesx.MODIFY.DEV	UPDATE
	$T RECVopts	jesx.MODIFY.RECVOPTS	CONTROL
	$T REDIRect	jesx.MODIFY.REDIRECT	CONTROL
	$T RMT	jesx.MODIFY.RMT	CONTROL
	$T SMFDEF	jesx.MODIFY.SMFDEF	CONTROL
	$T SOCKET	jesx.MODIFY.SOCKET	CONTROL
	$T SPOOL	jesx.MODIFY.SPOOL	CONTROL
	$T SPOOLDEF	jesx.MODIFY.SPOOLDEF	CONTROL
	$T SRVCLASS	jesx.MODIFY.SRVCLASS	CONTROL
	$T SSI	jesx.MODIFY.SSI	CONTROL
	$T STCCLASS	jesx.MODIFY.STCCLASS	CONTROL
	$T SUBMITLIB	jesx.MODIFY.SUBMITLIB	CONTROL
	$T SUBMITRDR	jesx.MODIFY.SUBMITRDR	CONTROL
	$T TPDEF	jesx.MODIFY.TPDEF	CONTROL
	$T TRACEDEF	jesx.MODIFY.TRACEDEF	CONTROL
	$T init stmt	jesx.MODIFY.init stmt	CONTROL
	$T TSUCLASS	jesx.MODIFY.TSUCLASS	CONTROL
$T J/B/T + O		jesx.MODIFY.JST*/BAT*/STC*/TSU*	
	$T J	jesx.MODIFY.BAT	UPDATE
	$T JOBQ	jesx.MODIFY.JST	UPDATE
	$T S	jesx.MODIFY.STC	UPDATE
	$T T	jesx.MODIFY.TSU	UPDATE
	$T O J	jesx.MODIFY.BATOUT	UPDATE
	$T O JOBQ	jesx.MODIFY.JSTOUT	UPDATE
	$T O S	jesx.MODIFY.STCOUT	UPDATE
	$T O T	jesx.MODIFY.TSUOUT	UPDATE
$VS*		jesx.VS	CONTROL
	$VS*	jesx.VS	CONTROL
$Z *		jesx.HALT.*	CONTROL
	$Z A	jesx.HALT.AUTOCMD	CONTROL

Command	Command option	RACF profile	Access level
	$Z I	jesx.HALT.INITIATOR	CONTROL
	$Z OFFLOADn	jesx.HALT.DEV	UPDATE
	$Z SPOOL	jesx.HALT.SPOOL	CONTROL
	$Z device	jesx.HALT.DEV	UPDATE
$ZAPJOB		jesx.ZAP.JOB	CONTROL
	$ZAPJOB	jesx.ZAP.JOB	CONTROL

Then, assign RACF permissions by using the **`RACF PERMIT`** command to certain user groups in your installation, as shown in the following profiles:

- System engineers privileged (for example, z/OS and JES2)
- System engineers from other product (for example, IBM Db2® IMS IBM CICS®)
- In-house operators
- Offshore operators
- Print operators
- System automation tasks, functions, or jobs

> **Attention:** The permissions that are given to the RACF profiles must match the profile that is given to the matching SDSF and EJES profiles.

6.9.3 SDSF and EJES considerations

The migration of SDSF and EJES security profiles is a simple process. You must add all RACF profiles of commands or panels that do not exist under JES3.

If you use the REXX interface of one of these third-party products (SDSF or EJES), you must consider the possibility of changing this use if you use fields that might no longer exist or were changed.

6.10 Migrating your printer

All printers that are defined in JES3 that use PSF must be migrated to JES2. Under JES3, you assign any printer name that you want if it meets the JES3 criteria.

Within JES2, all printers include a prefix in their name that is called PRT, followed by a four-digit number. Two different JES3 printer definitions from the customer project are shown in Example 6-26. In this example, Printer B433 and B439 turned on the separator page and the burst mode.

Example 6-26 JES3 Printer definitions

```
DEVICE,MODE=FSS,DTYPE=PRTAFP1,PM=(LINE,PAGE),FSSNAME=PSFGRP4A,
JNAME=B433,HEADER=YES,BURST=YES,DGROUP=PSFGRP4A,
JUNIT=(,*ALL,UR,ON),
CHARS=(YES,SC12),PAGELIM=0+,CKPNTPG=3,DYNAMIC=YES,
WS=(D,P,CL,F,L,C,PM,U),WC=2,FORMS=(h)
**-----------------------------------------------------------------
```

```
DEVICE,MODE=FSS,DTYPE=PRTAFP1,PM=(LINE,PAGE),FSSNAME=PSFGRP4A,
JNAME=B439,HEADER=NO,BURST=NO,DGROUP=PSFGRP4A,
JUNIT=(,*ALL,UR,ON),
CHARS=(YES,SC12),PAGELIM=0+,CKPNTPG=3,DYNAMIC=YES,
WS=(D,P,CL,F,L,C,PM,U),WC=2,FORMS=( )
```

The JES2 equivalent definitions for the printer are shown in Example 6-27. The printer name changed, as listed in Table 6-19.

Table 6-19 Comparison of JES printer names

JES3 printer name	JES2 printer name	JES2 writer name
B433	PRT1433	B433
B439	PRT1439	B439

The change in the printer name affects your installation. Consider the following points:

- Change printer permissions inside RACF if security is needed according to your new printer names PRT*. For more information about security definitions, see 6.9, “Migrating security” on page 157.
- Adjust all console commands or REXX that use native JES commands to the new printer names. Start, Stop, Modify, Forward, and Backward commands are targeted to the JES2 real printer name.

> **Attention:** The name of the JES2 writer name and route destination is set to the original JES3 printer name to be compatible with your applications. For more information, see the R and WRITER option in the JES2 printer definition and Example 6-27.

Example 6-27 JES2 printer definitions

```
/*-------------------------------------------------------------------*/
/*   PRINTERS FOR FSS GROUP PSFGRP4A                                 */
/*-------------------------------------------------------------------*/
PRT(1433) FSS=PSFGRP4A,R=B433,WRITER=B433,B=Y,SEP=Y
PRT(1439) FSS=PSFGRP4A,R=B439,WRITER=B439
/*-------------------------------------------------------------------*/
PRT(*)   CLASS=2,            /* DEFAULT CLASS FOR PRINT CENTER       */
         START=NO,           /* PRT1 COMES UP DRAINED                */
         PRMODE=(LINE,PAGE), /* PROCESS MODE                         */
         MODE=FSS,           /* WHETHER PRT IS STARTED UNDER         */
         WS=(Q,R/F,PRM,LIM,W,C,T,P), /* WORK SELECT. CRITERIA        */
         NPRO=90,                    /* PRINT TIMEOUT                */
         FORMS=(3820),       /* DEFAULT FORM TO PROCESS              */
         SEPDS=NO,           /* DEFAULT NO SEP PAGE                  */
         SEP=NO,             /* DEFAULT NO SEP PAGE                  */
         BURST=NO            /* DEFAULT NO BURST MODE                */
/*-------------------------------------------------------------------*/
```

6.10.1 FSS address spaces

The corresponding printer definitions in your FSS PSF started tasks must be changed slightly. The only one change that must be done is to change your printer names according to the new JES2 printer name. The appropriate JES3 FSS definition from printer B433 is shown in Example 6-28.

Example 6-28 JES3 FSS printer definition

```
//B433     CNTL
//B433     PRINTDEV FONTDD=*.FONT28, /* FONT     LIBRARY DD          */
//          OVLYDD=*.OLAY01,         /* OVERLAY  LIBRARY DD          */
//          PSEGDD=*.PSEG01,         /* SEGMENT  LIBRARY DD          */
//          PDEFDD=*.PDEF01,         /* PAGEDEF  LIBRARY DD          */
//          FDEFDD=*.FDEF01,         /* FORMDEF  LIBRARY DD          */
//          JOBHDR=*.JOBHDR,         /* JOB HEADER SEPARATOR OUTPUT  */
//          JOBTRLR=*.JOBTLR,        /* JOB TRAILER SEPARATOR OUTPUT */
//          DSHDR=*.DSHDR,           /* DATA SET HEADER SEPARATOR    */
//          MESSAGE=*.MSGDS,         /* MESSAGE DATA SET OUTPUT      */
//          PAGEDEF=A4H08,           /* DEVICE PAGEDEF DEFAULT       */
//          FORMDEF=EFA4,            /* DEVICE FORMDEF DEFAULT       */
//          CHARS=SC12,              /*                          @H1C*/
//          PIMSG=YES,               /* ACCUMULATE DATA SET MESSAGES */
//          TRACE=NO,                /* BUILD INTERNAL TRACE ENTRIES */
//          FAILURE=WCONNECT,        /* PSF ACTION ON PRINTER FAILURE*/
//          CONNINTV=86400,          /* jes connect interval time 1D */
//          TIMEOUT=REDRIVE,         /* PSF ACTION ON TIMEOUT        */
//          DISCINTV=0,              /* DISCONNECT INTERVAL IN SECOND*/
//          IPADDR='XXXXXXXXXX',     /* IP-ADDR                      */
//          PORTNO=4711              /* PORTNO                       */
//B433      ENDCNTL
```

The JES2 equivalent to the JES3 definition of printer B433 is shown in Example 6-29. The JES2 printer name that is used is based on your company's rules.

Example 6-29 JES2 FSS printer definition

```
//PRT1433  CNTL
//PRT1433  PRINTDEV FONTDD=*.FONT28, /* FONT     LIBRARY DD          */
//          OVLYDD=*.OLAY01,         /* OVERLAY  LIBRARY DD          */
//          PSEGDD=*.PSEG01,         /* SEGMENT  LIBRARY DD          */
//          PDEFDD=*.PDEF01,         /* PAGEDEF  LIBRARY DD          */
//          FDEFDD=*.FDEF01,         /* FORMDEF  LIBRARY DD          */
//          JOBHDR=*.JOBHDR,         /* JOB HEADER SEPARATOR OUTPUT  */
//          JOBTRLR=*.JOBTLR,        /* JOB TRAILER SEPARATOR OUTPUT */
//          DSHDR=*.DSHDR,           /* DATA SET HEADER SEPARATOR    */
//          MESSAGE=*.MSGDS,         /* MESSAGE DATA SET OUTPUT      */
//          PAGEDEF=A4H08,           /* DEVICE PAGEDEF DEFAULT       */
//          FORMDEF=EFA4,            /* DEVICE FORMDEF DEFAULT       */
//          CHARS=SC12,              /*                          @H1C*/
//          PIMSG=YES,               /* ACCUMULATE DATA SET MESSAGES */
//          TRACE=NO,                /* BUILD INTERNAL TRACE ENTRIES */
//          FAILURE=WCONNECT,        /* PSF ACTION ON PRINTER FAILURE*/
//          CONNINTV=86400,          /* jes connect interval time 1D */
//          TIMEOUT=REDRIVE,         /* PSF ACTION ON TIMEOUT        */
//          DISCINTV=0,              /* DISCONNECT INTERVAL IN SECOND*/
//         IPADDR='XXXXXXXXXX',     /* IP-ADDR                         */
//         PORTNO=4711              /* PORTNO                          */
//PRT1433   ENDCNTL
```

Tip: You can place JES2 and JES3 printer definitions in one FSS start procedure if the lines are not exceeded. This configuration prepares your FSS procedure for both JES versions. Alternatively, you can create a separate PROCLIB for all JES2 FSS procedures and replace them during the migration process.

6.11 Performance experience

This section provides information about general performance comparisons between JES3 and JES2. We also provide recommendations for the JES2 system layout.

6.11.1 CPU use comparison

In preparation for a JES3 migration project, no detailed information is available about the expected CPU use of a JES2 MAS versus JES3 complex. Because the CPU consumption is a key factor on IBM Z, we provide more information based on z13® hardware for the customer project.

The CPU consumption of all JES STCs that are running in an eight-way UAT sysplex during the migration time frame is shown in Figure 6-15. The chart also shows non-JES dependent workloads in user address spaces (the values are in MSU). The migration from JES3 to JES2 occurred on November 29, 2015 from 10:00 AM - 1:00 PM.

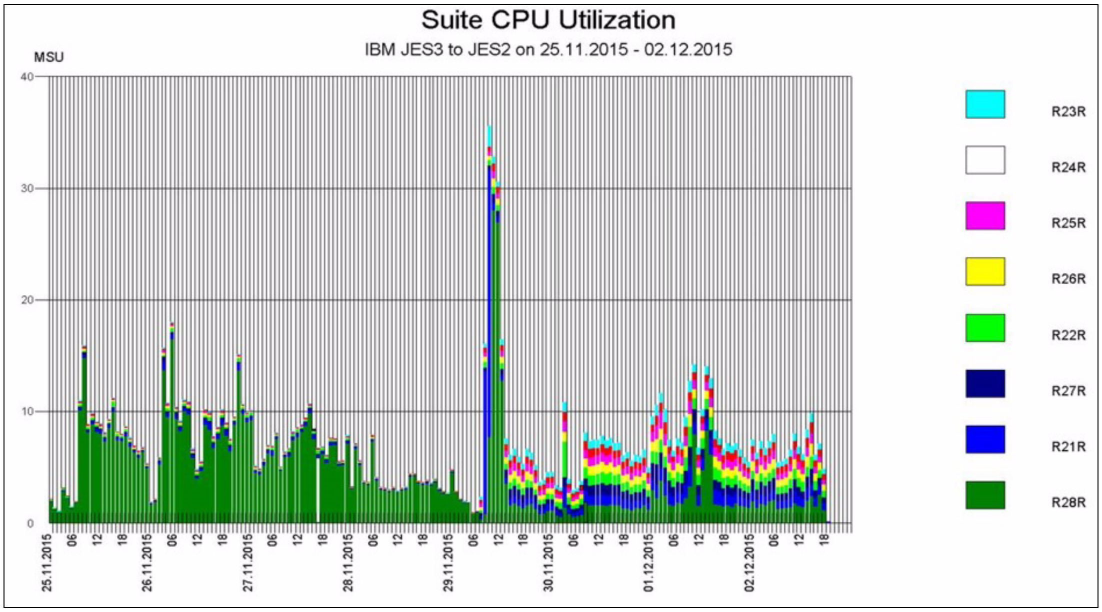


Figure 6-15 JES2/JES3 CPU consumption

Before the migration time, you see most of the CPU consumption is coming from system R28R, where the JES3 GLOBAL was stored. All of the JES3 LOCALS CPU consumption is low.

During the migration, the CPU consumption is higher because of all of the migration tasks that must be completed. After the migration to JES2, you see a more balanced CPU consumption across the sysplex because JES2s are independent of each other and have no GLOBAL equivalent, as JES3 does. This result might vary in your environment because of different workloads.

Conclusion

The total amount of CPU consumption per sysplex under JES2 is slightly lower compared to JES3. The reason for this result is that some of conversion is done in the user's address space instead of the JES address space. (This extra CPU workload is not considered in Figure 6-15.)

> **Note:** You can configure your JES2 to move Converter and Interpreter to a separate JES address space that is named JES2CIxx by using the INTERPRET=JES initialization option. This process also moves the workload from the user's address space to JES2 and provides a better comparison between both JES versions.

6.11.2 Dynamic checkpoint

Within JES2, a single resource that is named CHECKPOINT is available. This resource is used to share JES2-relevant information across all participating JES2 MAS members in the sysplex.

Each of the participating JES2 members includes a dedicated defined time to access the checkpoint. The checkpoint is shared by MAS members in a time-sliced manner.

Each member gets a lock on the checkpoint data set, reads the changes that were made by other members, processes the queues, writes updated control blocks back to the checkpoint, and releases the lock. It then waits before trying to access the checkpoint again.

We recommend that you consider the use of dynamic checkpointing with this release of z/OS. Using dynamic checkpointing brings your JES2 MAS in position to manage the HOLD and DORMANCY that is based on the JES2 workload on each of the participating systems in your MAS.

In a typical customer situation, you do not know in detail from where the JES2 workload is coming. If you know the origin of the workload, it can be changed rapidly because of moving subsystems to another system or workload considerations. An adjustment of the HOLD and DORMANCY values is required whether you know the origin of the workload.

> **Information:** In the customer environment, we saw a huge reduction of SPOOL delays in JES2 related workload.

To activate the support, code the JES2 initialization parameter MASDEF CYCLEMGT=AUTO, as shown in Example 6-7 on page 124. If JES2 is running, you can also enable the function dynamically by using the JES2 command **`$TMASDEF,CYCLEMGT=AUTO`**.

6.12 Hints and tips

In this section, we summarize the issues that we experienced during the JES2 migration process. This experience can vary on your site and has no claim to completeness.

6.12.1 JCL errors

After replacing JES3 with JES2, we observed several JCL errors in production BATCH processing. The root cause was the different handling of the JCL DD DLM option. The customer's production JCL included the jobs that are shown in Example 6-30.

Example 6-30 JCL DLM example

```
//SYSIN DD *,DLM=$$
first line
second line
//third line
$$
```

JES3 DLM handling

Under JES3, you can use the DLM option with DD * or DD DATA. In both cases, the SYSIN records are read until the characters that are defined in the DLM option appear.

JES2 DLM handling

JES2 handles DLM differently from JES3. If you code a DLM character in your JCL with DD *, the input stream is read until the DLM character or // appears. The differences between both JES versions when DD *,DLM JCL is used are listed in Table 6-20.

Table 6-20 Comparison of DLM usage

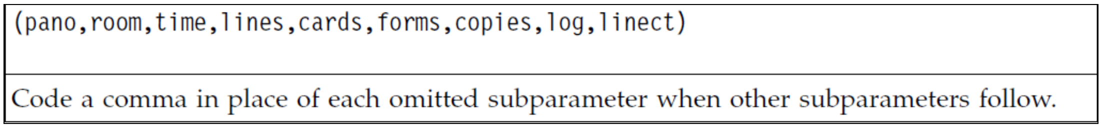
DLM statement	JES3	JES2
DD *,DLM=$$	Read the data until $$ appears	Read the data until $$ or // appears
DD DATA,DLM=$$	Read the data until $$ appears	Read the data until $$ appears

The solution for this issue is to migrate all of your JCL by using DD *,DLM= to DD DATA,DLM=.

6.12.2 S722 abends in JCL

Some customer's jobs were abending with S722, which means that the number of lines that is produced by these jobs exceeded a certain limit. A customer's limit was set to 16 message I/O (MIO) lines, as shown in Example 6-8 on page 127. Even so, some customer jobs were abending after only one line was produced.

The root cause was determined to be the different handling of the JCL accounting field of the job card. The handling of the accounting field under JES2 is shown in Figure 6-16.



```
(pano,room,time,lines,cards,forms,copies,log,linect)
```

Code a comma in place of each omitted subparameter when other subparameters follow.

Figure 6-16 Structure of JCL accounting field

In some customer jobs, the following similar job card was used:

```
//JOBA JOB (ITSO,SYSLAB,LUTZ,1),CLASS=M
```

According to the description of the account field that is shown in Figure 6-16 on page 172, the fourth value in the accounting field is honored as the maximum number of lines your job can produce. Therefore, the job is canceled after producing one line with the S722 abend. This behavior is the standard behavior of JES2. To eliminate this behavior, use the JES2 initialization parameter JOBDEF ACCTFLD=IGNORE.

6.12.3 Lost printer names after transfer

In the customer environment, print output is collected from all sysplexes in one sysplex where the office printers are connected. For this purpose, we used the manual transfer that is controlled by the system automation because JES2 does not provide such functions. For more information, see 6.8, “Migrating system automation” on page 153.

6.12.4 Monitoring default job class A

After migrating to JES2, monitor your set default JES2 job class (the standard is A). Because many people make mistakes during the conversion process, their user JCL can include many incorrect or missing job classes (see Example 6-31). The job card for JOBA does not contain any job class definition. It is possible that the user removed the //*MAIN CLASS= statement without moving that information to the job card.

Example 6-31 Missing JES2 job class definitions

```
//JOBA JOB (ACCT,ITSO,LUTZ),MSGCLASS=X,TIME=1440
//EXEC DD PGM=IEFBR14
//
//JOBB JOB (ACCT,ITSO,LUTZ),MSGCLASS=X,TIME=1440
//*MAIN CLASS=M
//EXEC DD PGM=IEFBR14
//
```

The second job JOBB appears not to be converted. The old JES3 //*MAIN CLASS statement still exists and is ignored by JES2 because it is only a comment.

In both situations, JOBA and JOBB are assigned to the default job class that is defined by JES2. This situation caused many delays in processing the default job class. For a brief overview of how many jobs are waiting to be run, use the **$DQ,Q=XEQ** JES2 operator command, as shown in Example 6-32.

Example 6-32 Sample output $DQ,Q=XEQ

```
$DQ,Q=XEQ
$HASP647     11 CNV          SYSA
$HASP647    400 XEQ A        SYSA
$HASP647      7 XEQ M        SYSA
$HASP647      5 XEQ M1       SYSA
$HASP647      1 XEQ P1       SYSA
$HASP647      1 XEQ P3       SYSA
$HASP647      5 XEQ S0       SYSA
$HASP647      5 XEQ S1       SYSA
$HASP647      1 XEQ S2       SYSA
```

As shown in Example 6-32, 400 jobs are waiting for running in JES2 default job class A because many of those jobs were misplaced because of incorrect job class information.

6.12.5 Monitor JES2 resources

JES2 uses many resources as listed in Table 6-21.

Table 6-21 JES2 resource list

Resource	Description	Set by	Scope
BERT	Block Extension reuse tables	BERTNUM on CKPTSPACE	SYS
BSCB	Bisynchronous buffers	BSCBUF on TPDEF	SYS
BUFX	Extended logical buffers	EXTBUF on BUFDEF	SYS
CKVR	Checkpoint versions	NUMBER on the CKPTDEF statement	SYS
CMBS	Console message buffers	BUFNUM on the CONDEF statement	SYS
CMDS	Console message buffers used for JES2 commands	CMDNUM on the CONDEF statement	SYS
ICES	IBM VTAM® sessions	SESSIONS on the TPDEF statement	SYS
LBUF	Logical buffers	BELOWBUF on the BUFDEF statement	SYS
JNUM	Job numbers	RANGE on the JOBDEF statement	MAS
JQES	Job queue elements	JOBNUM on the JOBDEF statement	MAS
JOES	Job output elements	JOENUM on the OUTDEF statement	MAS
NHBS	NJE header/trailer buffers	HDRBUF on the NJEDEF statement	SYS
SMFB	System management facility buffers	BUFNUM on the SMFDEF statement	SYS
TBUF	TCP/IP buffer data space	No external settings defined	SYS
TGS	SPOOL space/track groups	TGSPACE=(MAX=) on the SPOOLDEF statement	MAS
TTAB	Trace tables	TABLES on the TRACEDEF statement	SYS
VTMB	VTAM buffers	SNABUF on TPDEF	SYS
ZJC	Zone Job Container	ZJCNUM on GROUPDEF statement	MAS

These resources can be set according to your needs in the JES2 initialization PARMLIB member. In addition to the value of each resource, you can add a threshold value when you are notified that the value exceeds a previously defined threshold. In such cases, a generic $HASP050 message appears that indicates the resource type that caused the issue.

If a message appears, system operations often are not yet affected. The message that is coming from the job output elements resource is shown in Example 6-33. Therefore, the number of jobs in the JES2 output queue exceed 80% of total defined maximum.

Example 6-33 $HASP050 example

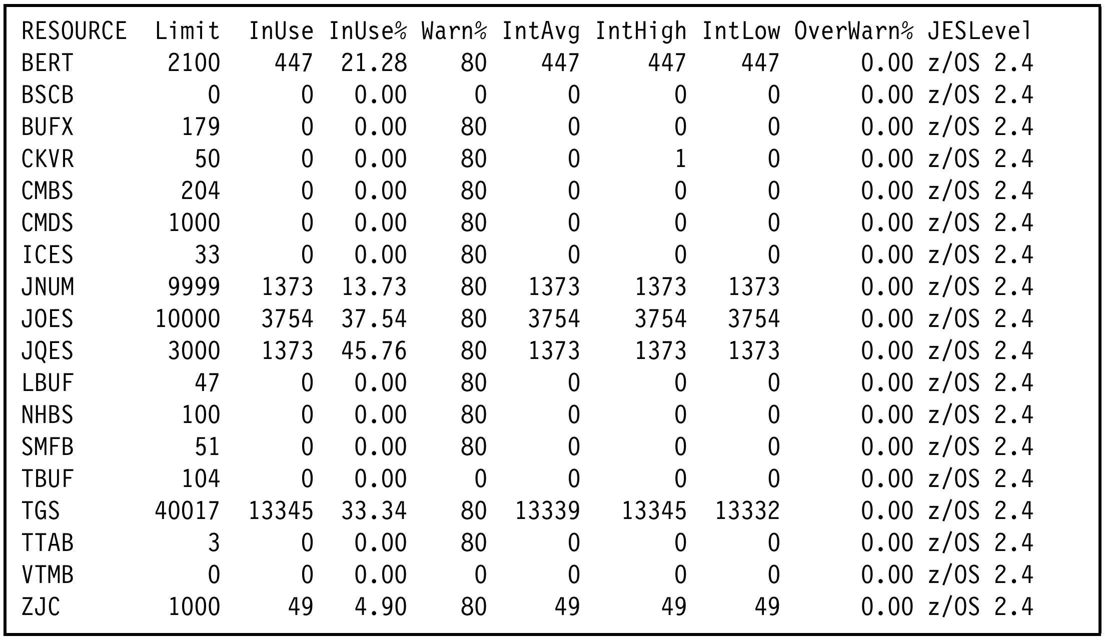
```
$HASP050 JES2 RESOURCE SHORTAGE OF JOES – 80% UTILIZATION REACHED
```

This message is a warning that the threshold for that particular resource was reached. Investigate the root cause of that message and take one of the following actions to solve the situation to avoid future problems:

- Run the **$OQ** command to release held output.
- Purge unneeded output.
- Make unprocessed output eligible for selection by changing printer characteristics.

If the messages appear too often, consider increasing the value of that resource.

An overview of all JES2 resources in a sample production system is shown in Figure 6-17. All values are defined in such a way that enough space still exists for unplanned actions.



```
RESOURCE  Limit  InUse InUse% Warn% IntAvg IntHigh IntLow OverWarn% JESLevel
BERT       2100    447  21.28    80    447     447    447      0.00 z/OS 2.4
BSCB          0      0   0.00     0      0       0      0      0.00 z/OS 2.4
BUFX        179      0   0.00    80      0       0      0      0.00 z/OS 2.4
CKVR         50      0   0.00    80      0       1      0      0.00 z/OS 2.4
CMBS        204      0   0.00    80      0       0      0      0.00 z/OS 2.4
CMDS       1000      0   0.00    80      0       0      0      0.00 z/OS 2.4
ICES         33      0   0.00    80      0       0      0      0.00 z/OS 2.4
JNUM       9999   1373  13.73    80   1373    1373   1373      0.00 z/OS 2.4
JOES      10000   3754  37.54    80   3754    3754   3754      0.00 z/OS 2.4
JQES       3000   1373  45.76    80   1373    1373   1373      0.00 z/OS 2.4
LBUF         47      0   0.00    80      0       0      0      0.00 z/OS 2.4
NHBS        100      0   0.00    80      0       0      0      0.00 z/OS 2.4
SMFB         51      0   0.00    80      0       0      0      0.00 z/OS 2.4
TBUF        104      0   0.00     0      0       0      0      0.00 z/OS 2.4
TGS       40017  13345  33.34    80  13339   13345  13332      0.00 z/OS 2.4
TTAB          3      0   0.00    80      0       0      0      0.00 z/OS 2.4
VTMB          0      0   0.00     0      0       0      0      0.00 z/OS 2.4
ZJC        1000     49   4.90    80     49      49     49      0.00 z/OS 2.4
```

Figure 6-17 JES2 resource display

6.12.6 Modifying JES3 OUTSERV

During the final migration to JES2, we decided to move files from selected JES3 SPOOL classes to JES2. During the transfer, we faced an issue that some JCL outputs were split in two or more pieces on the JES2 system. Therefore, the outputs were no longer all in one output file.

This issue affected of the output that were in the SPOOL files that were created with an SVC99 on the JES3 site. This issue occurred when the application used SVC99 for creating JES2 SPOOL data set; for example, memory dumps.

The solution was to code SNAGROUP=YES in the JES3 OUTSERV statement, as shown in Example 6-34.

Example 6-34 Sample JES3 OUTSERV

```
OUTSERV,CARRIAGE=7827,FORMS=7817,WS=(D,T,F,P,C,U,FL,CM,SS,CL,L),
WC=(0,1,2,3,4,5,6,7,9,A,B,D,F,G,H,I,J,K,M,N,P,Q,S,T,U,V,W,X,Y,Z),
THRESHLD=25000,TRAIN=H11,FLASH=NONE,OUTSVFCT=5,SNAGROUP=YES,
CHARS=(SC12),STACKER=C,CB=N
```

6.12.7 NJE performance

Based on the decision to move selected JES3 output classes to JES2, we recommend defining the maximum number of parallel NJE sender and receiver channels to get the maximum performance and reduce migration time. The appropriate NJEDEF statement with the SRNUM and STNUM option set to 4 is shown in Example 6-8 on page 127. By using this configuration, you can transfer four SYSOUT data sets in parallel.

> **Attention:** Do not forget to configure the pairing JES3 NJE server to four lines by using the OUTTRANS= parameter on the NJERMT JES3 initialization statement.

The JES2 and JES3 commands that are used to change to number of sysoutt channels is shown in Example 6-35.

Example 6-35 NJE modification

```
JES2 $TLINE(<your line number>),SRNUM=4,STNUM=4
JES3 *F,NJE,N=<your system name>,OR=4,OT=4
```

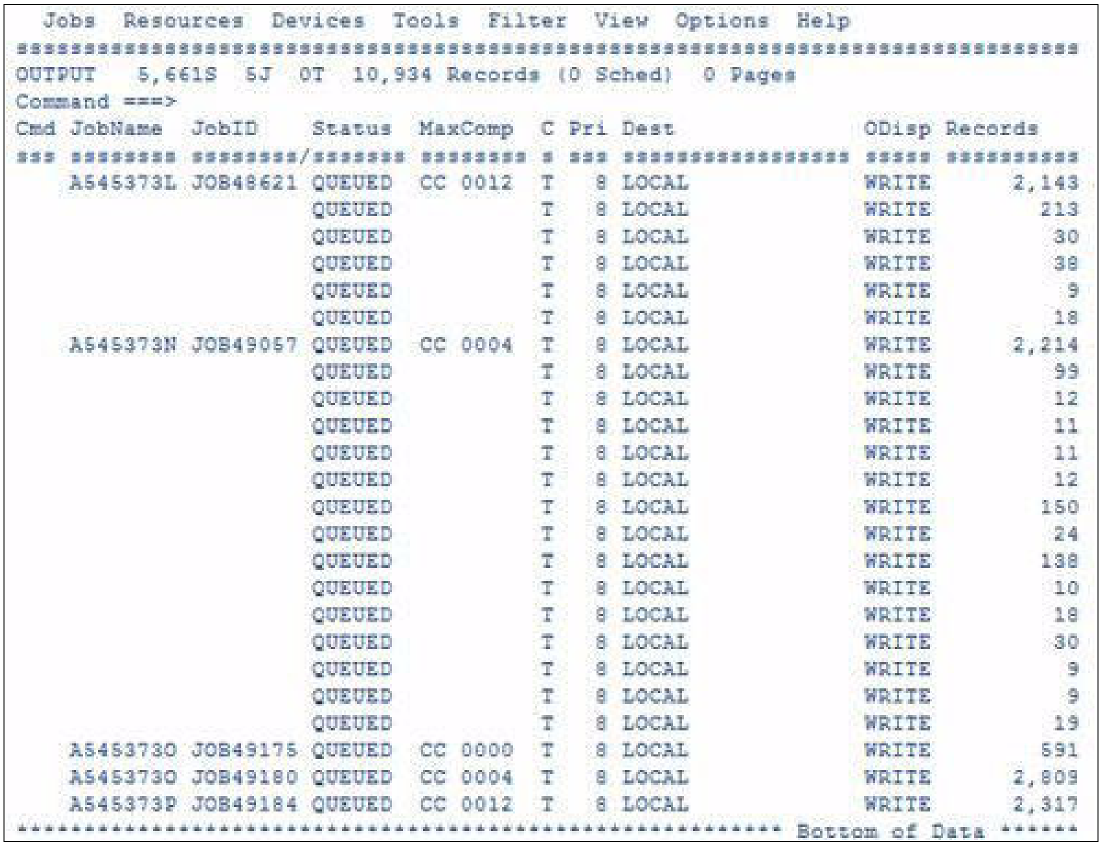
6.12.8 REXX SPIN

During the first business day, the customer saw a high use of JES2 job output elements (JOEs). The situation is brought to the customer’s attention when the following JES2 message appeared:

```
$HASP050 JES2 RESOURCE SHORTAGE OF JOES – 80% UTILIZATION REACHED
```

For more information, see 6.12.5, “Monitor JES2 resources” on page 174.

Two jobs that have more than one output data set allocated are shown in Figure 6-18 on page 177. Each job acquires one JOE.



```
  Jobs  Resources  Devices  Tools  Filter  View  Options  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
OUTPUT   5,661S  5J  0T  10,934 Records (0 Sched)  0 Pages
Command ===>
Cmd JobName  JobID    Status  MaxComp  C Pri Dest              ODisp Records
sss ssssssss ssssssss/sssssss ssssssss s sss sssssssssssssssss sssss ssssssssss
    A545373L JOB48621 QUEUED  CC 0012  T   8 LOCAL             WRITE      2,143
                      QUEUED           T   8 LOCAL             WRITE        213
                      QUEUED           T   8 LOCAL             WRITE         30
                      QUEUED           T   8 LOCAL             WRITE         38
                      QUEUED           T   8 LOCAL             WRITE          9
                      QUEUED           T   8 LOCAL             WRITE         18
    A545373N JOB49067 QUEUED  CC 0004  T   8 LOCAL             WRITE      2,214
                      QUEUED           T   8 LOCAL             WRITE         99
                      QUEUED           T   8 LOCAL             WRITE         12
                      QUEUED           T   8 LOCAL             WRITE         11
                      QUEUED           T   8 LOCAL             WRITE         11
                      QUEUED           T   8 LOCAL             WRITE         12
                      QUEUED           T   8 LOCAL             WRITE        150
                      QUEUED           T   8 LOCAL             WRITE         24
                      QUEUED           T   8 LOCAL             WRITE        138
                      QUEUED           T   8 LOCAL             WRITE         10
                      QUEUED           T   8 LOCAL             WRITE         18
                      QUEUED           T   8 LOCAL             WRITE         30
                      QUEUED           T   8 LOCAL             WRITE          9
                      QUEUED           T   8 LOCAL             WRITE          9
                      QUEUED           T   8 LOCAL             WRITE         19
    A545373O JOB49175 QUEUED  CC 0000  T   8 LOCAL             WRITE        591
    A545373O JOB49180 QUEUED  CC 0004  T   8 LOCAL             WRITE      2,809
    A545373P JOB49184 QUEUED  CC 0012  T   8 LOCAL             WRITE      2,317
********************************************************* Bottom of Data ******
```

Figure 6-18 JES2 output queue

The root cause was the **TSO FREE** command. This command includes the default option SPIN(UNALLOC), which closes the data set and generates a JOE in JES2 SPOOL.

By using the SPIN(NO) option in the **FREE** command, the output data sets are not closed immediately. Instead, they are closed at the end of the job (REXX). Therefore, only one JOE in JES2 SPOOL is occupied per job. The differences in the commands are shown in the following examples:

- Old command: `FREE D(<DDNAME>)`
- New command: `FREE D(<DDNAME>) SPIN(NO)`

6.12.9 NJE parms for time differences

In the customer environment, we use a UAT sysplex that runs with a date in the future to verify new application programs. Therefore, we have a time difference between that sysplex and all of the other application programs that are running. To establish an NJE connection between systems with different times, use the NJEDEF TIMETOL=0 option during JES2 initialization.

A UAT sysplex runs with a time in the future. It was not possible to establish a connection to this sysplex from the system that was set to normal time. The UAT sysplex is reset to normal time every quarter. Then, connection problems occurred again because the remaining NJE nodes stored a later time stamp than the sysplex now used.

This behavior was not caused by NJE, but by the pathmanager functionality. To avoid this issue, turn off the path manager of JES2 by using the PATHMGR=NO option.

Attention: If you must use the PATHMGR=NO option, you must manually define all of your NJE network routes.

To establish a static NJE connection without the use of NJE path manager capability, you must manually define all network routes.

A sample NJE configuration with three systems that connect over TCP/IP is shown in Figure 6-19. With PATHMGR=YES, no other definitions to JES2 are necessary to connect nodes.

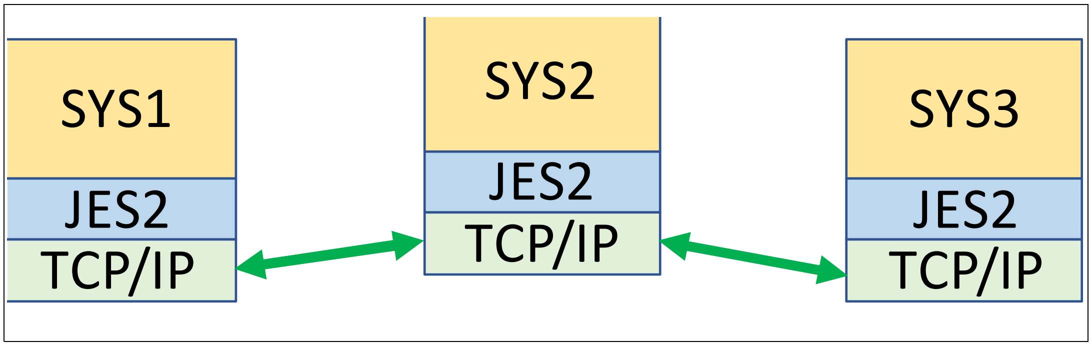


Figure 6-19 JES2 NJE configuration

If you are requested to use PATHMGR=NO, you must manually define the route from SYS1 to SYS3. The following statement must be placed in your JES2 PARMLIB configuration data set for system SYS1:

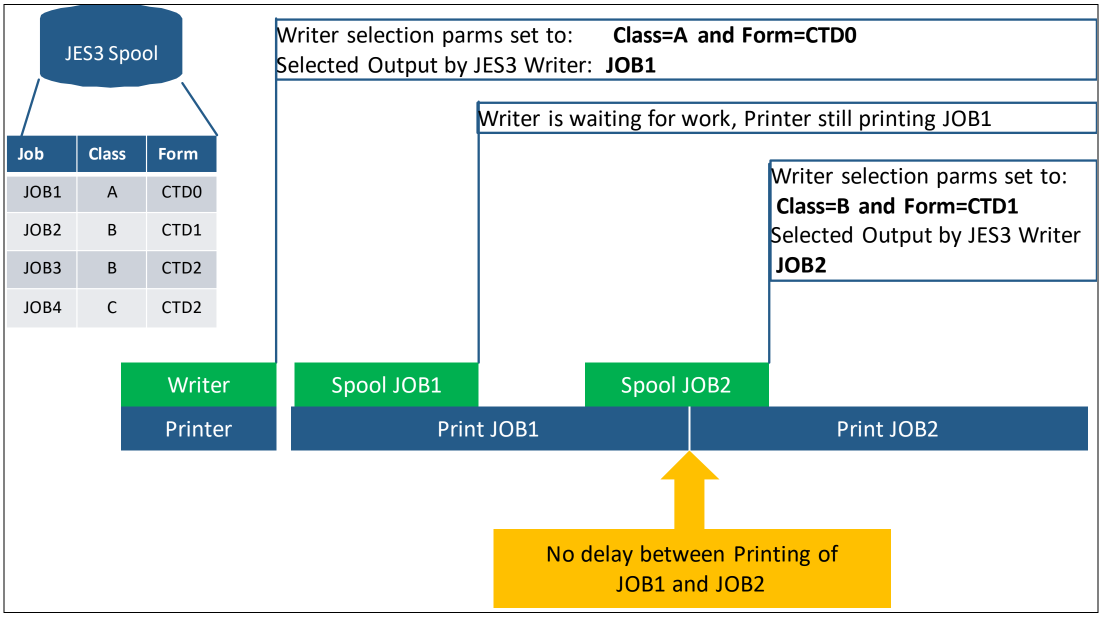
```
CONNECT PATHMGR=NO,NA=SYS2,NB=SYS3
```

This statement tells NJE on SYS1 that node SYS3 is connected or reachable over SYS2. On SYS3, you must define the route in the opposite manner, as shown in the following example:

```
CONNECT PATHMGR=NO,NA=SYS2,NB=SYS1
```

6.12.10 Print delays

For a customer’s high-performance print center within JES3, they can change their printer selection criteria while the printer was active and printing. This change prevents the printer from stopping. A printer that stops leads to another warm-up phase of the printer, which wastes approximately 50 blank pages. The print flow within JES3 is shown in Figure 6-20 on page 179.



Job	Class	Form
JOB1	A	CTD0
JOB2	B	CTD1
JOB3	B	CTD2
JOB4	C	CTD2

Writer selection parms set to: **Class=A and Form=CTD0**
Selected Output by JES3 Writer: **JOB1**

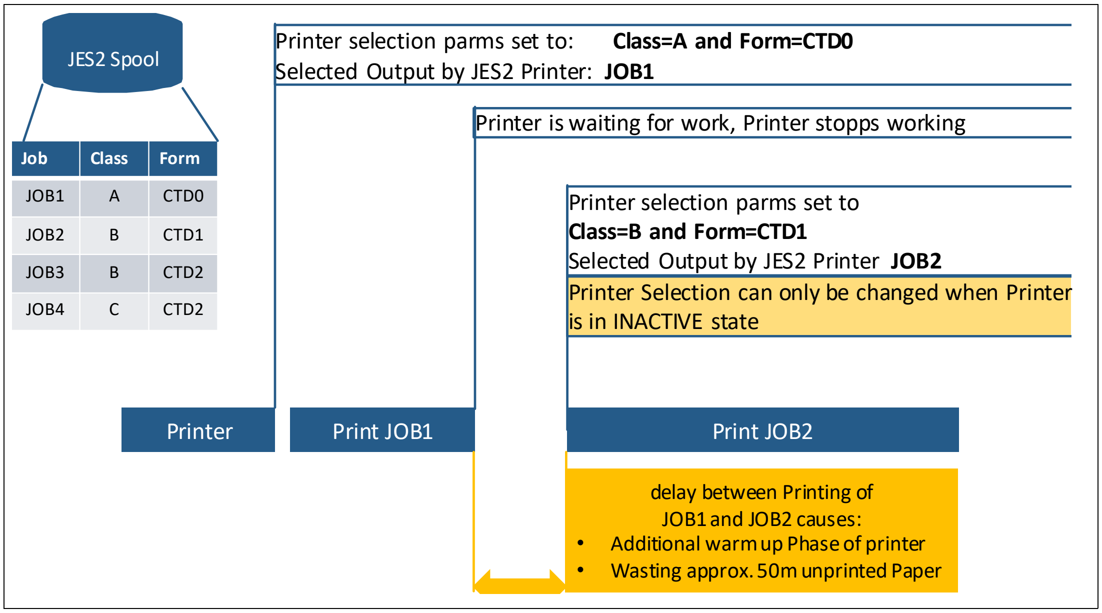
Writer is waiting for work, Printer still printing JOB1

Writer selection parms set to:
Class=B and Form=CTD1
Selected Output by JES3 Writer
JOB2

No delay between Printing of JOB1 and JOB2

Figure 6-20 JES3 Printing

Within JES2, you cannot change the printer selection criteria, such as sysout class and forms while the printer is active. How JES2 works with printers is shown in Figure 6-21. In this example, we start the JES2 printer for sysout class A and forms CTD0. The first job for processing is JOB1. The next waiting job to print JOB2 is coming from sysout class B and form CTD1. The printer must be inactive to change the printer's selection criteria.



Job	Class	Form
JOB1	A	CTD0
JOB2	B	CTD1
JOB3	B	CTD2
JOB4	C	CTD2

Printer selection parms set to: **Class=A and Form=CTD0**
Selected Output by JES2 Printer: **JOB1**

Printer is waiting for work, Printer stopps working

Printer selection parms set to
Class=B and Form=CTD1
Selected Output by JES2 Printer **JOB2**

Printer Selection can only be changed when Printer is in INACTIVE state

delay between Printing of JOB1 and JOB2 causes:

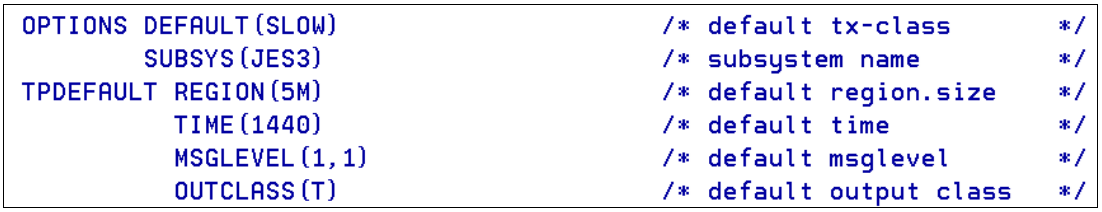
- Additional warm up Phase of printer
- Wasting approx. 50m unprinted Paper

Figure 6-21 JES2 Printing

Stopping the printer causes at least a waste of paper. To avoid this issue, we recommend starting your printer with parms to process more than one sysout class (up to eight are possible).

6.12.11 APPC abends

After restarting the systems with JES2, all APPC/ASCH address space failed. The problem occurred because of a hardcoded sub system declaration in customers ASCHPM00 member, as shown in Figure 6-22.



```
OPTIONS DEFAULT(SLOW)                         /* default tx-class        */
        SUBSYS(JES3)                          /* subsystem name          */
TPDEFAULT REGION(5M)                          /* default region.size     */
          TIME(1440)                          /* default time            */
          MSGLEVEL(1,1)                       /* default msglevel        */
          OUTCLASS(T)                         /* default output class    */
```

Figure 6-22 ASCHPM00 member

The solution was to remove that SUBSYS(JES3) statement. The primary subsystem is used if this option is omitted.

Note: It is suggested to scan all of your z/OS PARMLIBs for occurrences of the JES3 keyword to identify such mis-configurations in advance.

6.13 Ready to migrate

In this section, we describe how a JES3 migration can be done based on a customer experience. Our example is based on the following steps:

1. Prepare your sysplex.
2. Shut down JES3 Sysplex.
3. Restart Sysplex with JES2 MAS until TSO.
4. Prepare the NJE connection to JES2 MAS.
5. Start SPOOL migration.
6. Start JES2 tests and sample jobs.
7. Restart subsystems, such as Db2, IMS, and CICS.
8. Release your BATCH.

During the migration, all subject matter experts must be available to the control their subsystems and conduct tests after JES2 is activated.

Information: Any subject matter expert must confirm that their product is working with JES2 after migrating to the project.

6.13.1 Preparing your sysplex

First, create a saved copy of your IEFSSNxx member in your PARMLIB. This saved copy is used if you must go back to JES3.

Replace the primary subsystem JES3 with JES2 in your active IEFSSNxx member, as shown in Example 6-36 and Example 6-37.

Example 6-36 JES3 IEFSSNxx entry

```
SUBSYS SUBNAME(JES3)
  PRIMARY(YES) START(NO)
```

Example 6-37 JES2 IEFSSNxx entry

```
SUBSYS SUBNAME(JES2)
  PRIMARY(YES) START(NO)
```

Attention: With a primary JES2 subsystem, it is not possible to have a parallel JES3 secondary subsystem available. The SUBSYS SUBNAME(JES3) must be removed from the IEFSSNxx member.

The next step is to prepare all participating subsystem products, especially those that are close in contact with JES.

JES2 initialization

It is recommended to start JES2 in front of the migration with a JES2 cold start. This process can be easily done by defining JES2 as the secondary subsystem in parallel to JES3 as the primary.

Stop BATCH processing

All BATCH jobs outside of system engineering should be stopped. This process can be done by stopping all jobs that are coming from your BATCH control system and preventing the start of jobs by JES3.

Attention: This step should be started well in advance because some jobs might be running for a long time, especially in production. Contact your BATCH scheduling team for more information.

6.13.2 Shutting down JES3 sysplex

Now you can begin shutting down JES3 sysplex (all at once or individually). For safety reasons, it is better to leave one system up with JES3. In the customer case study that is shown in Figure 6-23 on page 182, a separate JES3 system was added to the sysplex for the following reasons:

- Transfer of JES3 SPOOL files to JES2 (if needed)
- To have a backup system available if:
  - You must check how a process was working under JES3 for comparison with JES2
  - To access the system if JES2 does not work

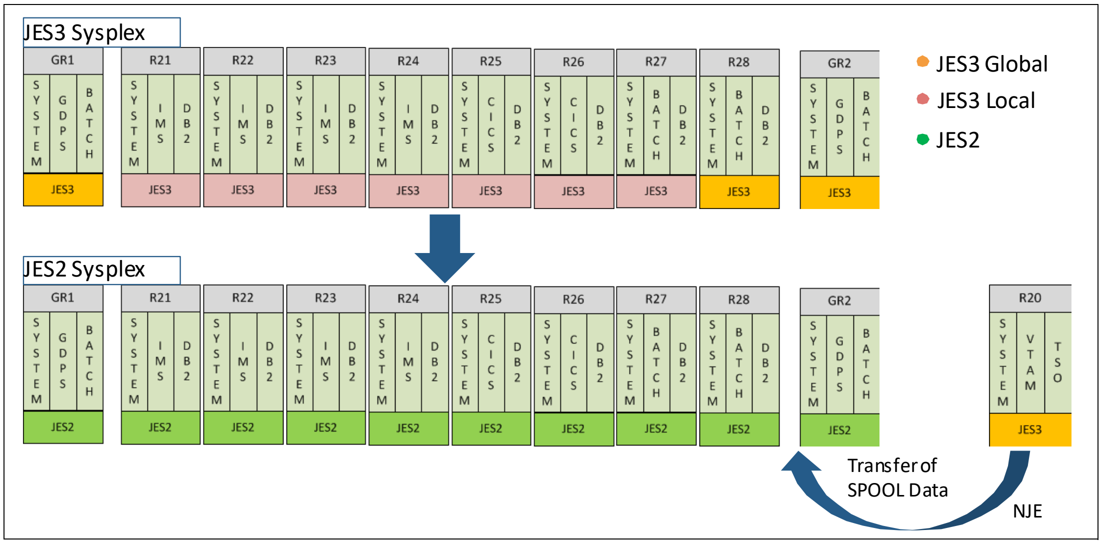


Figure 6-23 Migration to JES2

The extra system is part of the JES3 complex and becomes the new JES3 GLOBAL. That system was active for the next week after migration to JES2 because of the reasons that are described in this section.

6.13.3 Restarting sysplex with JES2 MAS until TSO

Now you can IPL all of the systems in your sysplex. Because the amount of time that a member can hold the checkpoint for and the time it waits before trying to reacquire the checkpoint is controlled by the **HOLD=** and **DORMANCY=** parameters on the **MASDEF** statement, prepare your JES2 MAS ahead of the migration, as described in 6.3, “JES2 system design” on page 124. The IPLs should be done up to TSO. Then, you can begin testing your JES2 infrastructure.

Refreshing automation table

The new automation table must be activated by using the **INGAMS REFRESH** automation command. The new table contains all of the new JES2 messages that must be processed and the new set procedures, if needed. This process can be done before the first activation of JES2.

Stop BATCH

To prevent unwanted jobs in your system, stop job processing by removing queue affinity from your JES2 job classes, as shown in Example 6-38.

Example 6-38 Stopping JES2 BATCH

```
$DJOBCLASS(<your job classes>),QAFF(ANY)=OFF
```

6.13.4 Preparing NJE connection to JES2 MAS

This step is optional for your migration. If you want to keep your JES3 SPOOL files and move them to JES2, you must establish an NJE connection between your new JES2 MAS and the remaining JES3 system.

Because the JES2 includes the same NJE node name as the JES3 before, you must change the node name for the JES3 system by completing the following steps:

1. Rename the JES3 home node definition that is shown in Example 6-39.
2. Add the JES2 partner node (the origin node name that JES2 now uses).

Example 6-39 Modified JES3 NJE configuration

```
NJERMT,NAME=SYS2,HOME=YES,MAXLINE=0,DEFCLASS=NO
NETSERV,NAME=NJENSRV,HOSTNAME=TCPSYS2
*-----------------------------------------------
NJERMT,NAME=SYS1,HOME=NO,TYPE=TCPIP
SOCKET,NAME=S1SYS1,NETSERV=NJENSRV,
HOSTNAME=NJE-SYS1,NODE=SYS1
```

Attention: After changing your JES3 INISH member, you need a JES3 hot start to pick up these changes.

The JES2 node also can be defined dynamically by using the JES3 commands that are shown in Example 6-40.

Example 6-40 Defining JES2 partner node

```
*F,NJE,ADD=SYS1,TYPE=TCPIP
*F,SOCKET,ADD=S1SYS1,HOSTNAME=TCPSYS1,NETSERV=NJENSRV,NODE=SYS1
```

Next, add the renamed node name to your JES2 MAS (see Example 6-41).

Example 6-41 JES NJE definitions

```
NODE(1)  NAME=SYS1           /* OWNNODE=1                            */
NETSRV1  SOCKET=SYS1
  SOCKET(SYS1) NODE=1,IPADDR=YOUR-ADRESS
/*-------------------------------------------------------------------*/
NODE(2) NAME=SYS2
  SOCKET(SYS2) NODE=2,IPADDR=ADRESS-SYS2,CONNECT=YES
```

Now you can establish the NJE connection between both systems by using the JES2 start command that is shown in Example 6-42.

Example 6-42 JES2 start connection to JES3

```
$SN,N=SYS2
$HASP000 OK
IAZ0543I NETSRV1 TCP/IP connection with IP Addr: TCPSYS2 Port: 175 Initiated
IAZ0543I NETSRV1 TCP/IP connection with IP Addr: TCPSYS2 Port: 175 Successful
```

The JES3 commands that are used for starting the JES2 node from the JES3 system are shown in Example 6-43.

Example 6-43 JES3 start connection to JES2

```
*S,TCP,SOCKET=S1SYS1
*S,TCP,NODE=S1SYS1
```

6.13.5 Starting SPOOL migration

First, identify the SPOOL content that must be transferred. This content depends on your company SPOOL sysout class definitions and can vary. To determine the amount of sysout you must transfer, you can use JES3 command that is shown in Example 6-44. This command shows you the number of SPOOL files in all HOLD and WTR sysout classes.

Example 6-44 Display JES3 sysout

```
*I U Q=HOLD,CL=?
IAT8131 CL=O, L=27586, PG=0, SR=27586, BY=2642348.
IAT8131 CL=L, L=1000000, PG=0, SR=1000000, BY=121360144.
IAT8131 CL=T, L=3464764, PG=0, SR=3464764, BY=305434192.
IAT8131 CL=Y, L=369, PG=0, SR=369, BY=36756.
IAT8119 NUMBER OF JOBS FOUND : 3579

*I U Q=WTR,CL=?
IAT8131 T=PRT, CL=A, L=199055, PG=0, SR=199055, BY=24222204.
IAT8119 NUMBER OF JOBS FOUND : 628
```

Information: To calculate the number of bytes that must be transferred and the time that is needed, do not use the number of bytes you see in SDSF. Instead, multiply the number of lines by the record length of 133 to calculate the number of bytes that must be transferred.

Some sample JES3 commands are shown in Example 6-45. The destination to your target system can be changed for all elements in sysout hold class X in this example.

Example 6-45 JES3 command for transfer

```
*F U Q=HOLD,CL=X,ND=<new Destination>,N=ALL
*F,U,Q=HOLD,CL=X,AGE>3,ND=<new Destination>,N=ALL
```

Attention: Before starting the SPOOL migration, ensure that the SNAGROUP=YES option is enabled in JES3 so that the output files are not split. For more information, see 6.12, “Hints and tips” on page 172.

6.13.6 JES2 test cases

After all your systems that are brought up with JES2, you can conduct basic system tests to verify that the system with JES2 is working as expected. Define your own test scenario that is based on your system environment by using the test case information that is listed in Table 6-22.

Table 6-22 Sample test cases after migration

Name	Description	Expected result
Check EDP	Jobs from EDP (end of Day Processing) might run	No JCL errors or abends caused by the migration
NJE Connectivity	Check active NJE lines	All defined NJE nodes active
JCL Test	Test JCL Job runs successfully	RC=0 on all test jobs
JES2 Access	Check JES2 modify commands for unauthorized users	Unauthorized users prevented from JES2 modify commands

Name	Description	Expected result
UserID and Password Test	Allocate new DSN with no permission. Submit job without password from secondary UID	RACF error S913
Test OPERATOR Security	► Stop / Start JES2 ► Restart Job ► Start / Stop BATCH Job,STC,TSU,JST ► Device Management	No security error
NJE Security	► Send print output to remote RZ ► Start job on remote RZ	No security error
Alarm Tests	Tests of alarm you can set with JES2	Alarm is showing on the monitor, email, and mobile phone
Batch OFF/ON	Check if BATCH can be stopped and started	All job classes must be off or on, based on their JES2 system affinity
Exit Test	Test all of your JES2 user modifications	JES2 exits works as designed
SYSLOG	Syslog works as expected	No problems identified
SWITCH_SYSLOG	Switch syslog data works as expected	No problems identified
SYSLOG_ARCHIVE	SYSLOG can be archived	No problems identified
Test Printing	► Print file from mainframe to remote printer ► Print from IMS	► Print file arrives in JES2 Spool ► File is printed

6.13.7 Restarting subsystem

After the basic system tests are completed successfully, you can consider restarting all subsystems. In our case study, we did observe any issue with all subsystems upon restart under JES2.

6.13.8 Releasing your BATCH

If all of the previous steps were successful, you can now consider releasing your BATCH jobs. This process includes enabling job submission in your BATCH controlling system and if it exists in your company, release system affinity in your JES2 job classes to allow jobs to run.

Attention: Carefully monitor JES2 default job class for misplaced jobs because of missing job class information.

6.13.9 Quitting your JES2 license

Quit your JES3 license at the end of the process. The cancellation period often is one month.

6.14 Project considerations

In this section, we describe some project-related line items that are non-technical. We start with the general project organization in a customer's environment.

6.14.1 Project organization

The project organization is based on the resources of people that are available for the project. Because most of the migration actions can be done in parallel, we suggest bringing as many people as you can to the project to lower the project time.

The project was separated in 10 subprojects, which were managed by a dedicated person. The overall project organization on the customer site is shown in Figure 6-24. The project contains 10 deliverables called Work products (WP). This number can vary in your environment based on your needs and your available head count.

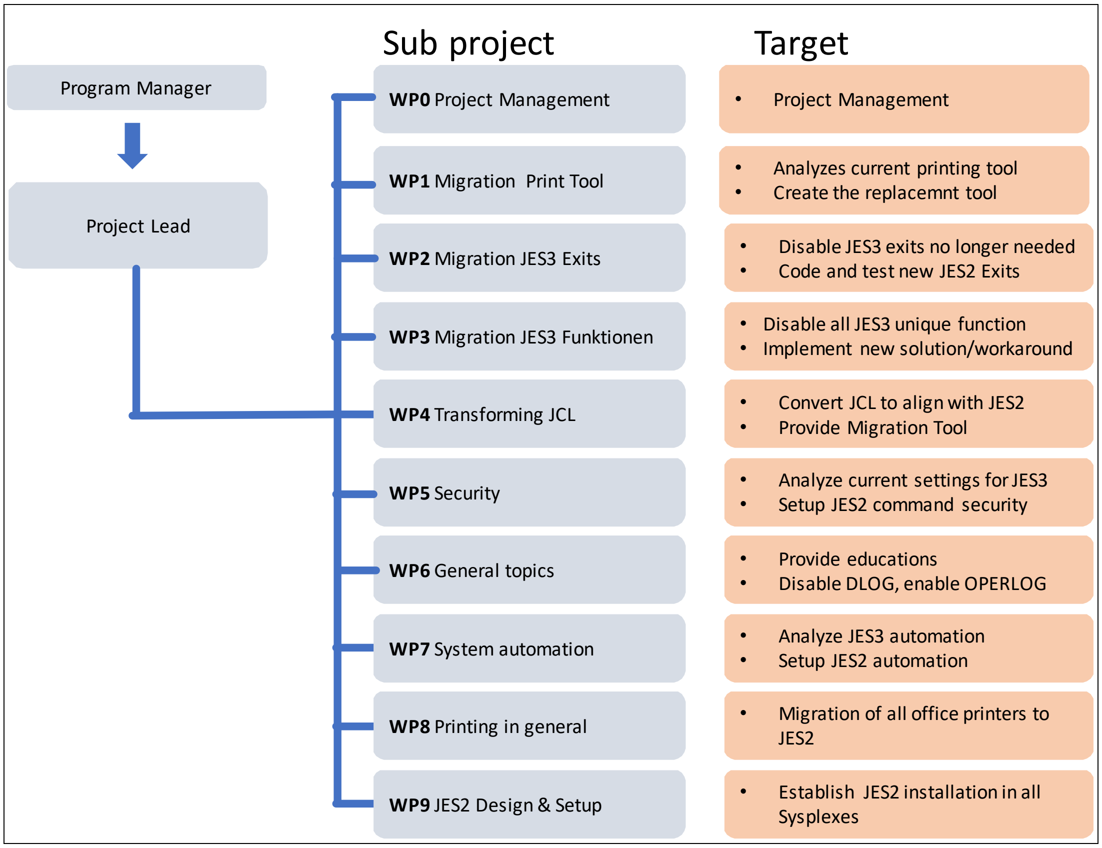

Sub project
Target
Program Manager
Project Lead
WP0 Project Management
• Project Management
WP1 Migration Print Tool
• Analyzes current printing tool
• Create the replacemnt tool
WP2 Migration JES3 Exits
• Disable JES3 exits no longer needed
• Code and test new JES2 Exits
WP3 Migration JES3 Funktionen
• Disable all JES3 unique function
• Implement new solution/workaround
WP4 Transforming JCL
• Convert JCL to align with JES2
• Provide Migration Tool
WP5 Security
• Analyze current settings for JES3
• Setup JES2 command security
WP6 General topics
• Provide educations
• Disable DLOG, enable OPERLOG
WP7 System automation
• Analyze JES3 automation
• Setup JES2 automation
WP8 Printing in general
• Migration of all office printers to JES2
WP9 JES2 Design & Setup
• Establish JES2 installation in all Sysplexes

Figure 6-24 Project organization example

WP0 Project management

In this subproject, many important non-technical activities should be covered, such as creating and maintaining:

- Project plan
- Stakeholder planning
- Communication plan
- Resource plan
- Required leave plan for all participants

The overall project plan should address all of the project needs in small pieces. This process helps you to monitor all line items and identify problems and delays early.

The plan that includes all affected stakeholders should be created to ensure that all persons that are affected by replacing JES3 are known. Based on that list, you start planning for education or communications to those stakeholders.

The communication plan is crucial for the project to improve the acceptance of the project for all stakeholders. All communication should be adjusted according to the following levels of the stakeholder:

- Information to the management (a 2-week cycle is sufficient)
- Information to the technical stakeholder on weekly basis
- All the other users, if needed

The next plan should cover all types of resource planning. The plan includes registering people that are needed for the project to ensure that they are available for the project. At the same time, carefully record all known planned absences to balance the available resources to avoid unwanted project delays.

WP1 and WP8 all kinds of printing

This work product varies in your installation and can be more simple on your site. One of the major targets at the customer site was the migration and reprogramming of a custom-made REXX printer control tool. This conversion took almost two months of investigation, programming, and testing of the new print solution.

The only one task that is left is the migration of the printer names to the JES2 printer naming convention.

WP2 JES3 exits

WP2 was established to cover all kind of things that needed to remove or convert all used JES3 exits in a customer's installation. It is recommended to create a list of all installed JES3 exits and rate what should be done we these exits, including the following options:

- Remove the exits and replace functionality by using other functions
- Remove the exit because it is no longer needed
- Convert the JES3 exit to JES2
- No action is required; JES3 exit does not exist under JES2 and is no longer needed after migration

In a customer's environment, the following exits were left that needed to reprogrammed:

- JES2 Exit 6 to adjust JCL with customer modifications
- JES2 Exit 23 is user for PSF to create a print header page

WP3 JES3 unique functions

This subproject covers all required actions that needed to replace JES3 functions, such as:

- Depended Job Control
- DEADLINE processing
- Disk reader

One significant task is to identify all users that use such functions. The following methods can be used to identify these users:

- Scan your production JCL libraries for the existence of JES3-specific JECL cards
- Keep your SYSLOG/OPERLOG data with at least one year and check for jobnames that are using such functions

> **Important:** After identifying all of the users, contact all of the job owners and prepare to migrate the jobs, if still needed. After JES2 is migrated, carefully monitor all migrated jobs.

WP4 migration of your JCL

The subproject intended to align all of your JCL and JECL to work with JES2 and JES3. The following areas were affected in the customer's environment:

- Migration of production BATCH processing
- Migration of technical jobs from infrastructure teams, mostly outside of BATCH management systems
- Migration of user defines JCL, which were in user datasets
- Identify and adjust all types of JCL generators that were used
- Identify job dependencies in terms of behavior of JES2 versus JES3

> **Important:** Our r experience shows that you can expect most problems after migration because many users made mistakes while migrating their JCL.

WP5 security

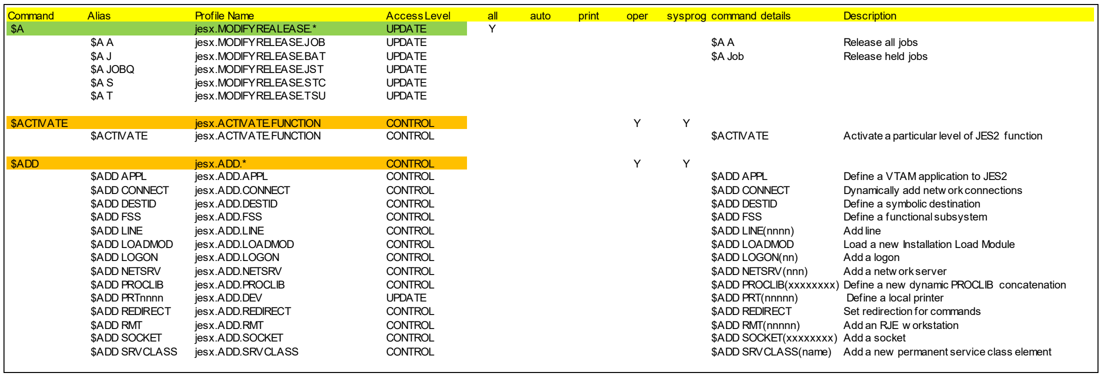
The purpose of subproject is to correctly set up the JES2 command security according to your JES3 settings. We created a list of all available JES2 commands and their corresponding RACF profile and assigned them to the users based on the JES3 settings.

Complete the following steps:

1. Create JES2 command security according to your JES3 definitions.
2. Define JES2 STC security definition, which also is part of WP9.
3. Define security for JES2 working data sets, such as SPOOL and CKPT.
4. Define (if necessary) new printer security definitions because the printer names are subject to change.

> **Important:** This work product can be tested in parallel to your JES3 in advance with the dependency to WP9 JES2 setup. If you use a running JES2 as a secondary subsystem, you can test your security setup.

A part of the table of JES2 commands and their permissions are shown in Figure 6-24. You must add columns for permissions that are based on your organization.



Command	Alias	Profile Name	Access Level	all	auto	print	oper	sysprog	command details	Description
$A		jesx.MODIFYREALEASE.*	UPDATE	Y						
	$A A	jesx.MODIFYRELEASE.JOB	UPDATE						$A A	Release all jobs
	$A J	jesx.MODIFYRELEASE.BAT	UPDATE						$A Job	Release held jobs
	$A JOBQ	jesx.MODIFYRELEASE.JST	UPDATE							
	$A S	jesx.MODIFYRELEASE.STC	UPDATE							
	$A T	jesx.MODIFYRELEASE.TSU	UPDATE							
$ACTIVATE		jesx.ACTIVATE.FUNCTION	CONTROL				Y	Y		
	$ACTIVATE	jesx.ACTIVATE.FUNCTION	CONTROL						$ACTIVATE	Activate a particular level of JES2 function
$ADD		jesx.ADD.*	CONTROL				Y	Y		
	$ADD APPL	jesx.ADD.APPL	CONTROL						$ADD APPL	Define a VTAM application to JES2
	$ADD CONNECT	jesx.ADD.CONNECT	CONTROL						$ADD CONNECT	Dynamically add network connections
	$ADD DESTID	jesx.ADD.DESTID	CONTROL						$ADD DESTID	Define a symbolic destination
	$ADD FSS	jesx.ADD.FSS	CONTROL						$ADD FSS	Define a functional subsystem
	$ADD LINE	jesx.ADD.LINE	CONTROL						$ADD LINE(nnnn)	Add line
	$ADD LOADMOD	jesx.ADD.LOADMOD	CONTROL						$ADD LOADMOD	Load a new Installation Load Module
	$ADD LOGON	jesx.ADD.LOGON	CONTROL						$ADD LOGON(nn)	Add a logon
	$ADD NETSRV	jesx.ADD.NETSRV	CONTROL						$ADD NETSRV(nnn)	Add a network server
	$ADD PROCLIB	jesx.ADD.PROCLIB	CONTROL						$ADD PROCLIB(xxxxxxxx)	Define a new dynamic PROCLIB concatenation
	$ADD PRTnnnn	jesx.ADD.DEV	UPDATE						$ADD PRT(nnnnn)	Define a local printer
	$ADD REDIRECT	jesx.ADD.REDIRECT	CONTROL						$ADD REDIRECT	Set redirection for commands
	$ADD RMT	jesx.ADD.RMT	CONTROL						$ADD RMT(nnnnn)	Add an RJE workstation
	$ADD SOCKET	jesx.ADD.SOCKET	CONTROL						$ADD SOCKET(xxxxxxxx)	Add a socket
	$ADD SRVCLASS	jesx.ADD.SRVCLASS	CONTROL						$ADD SRVCLASS(name)	Add a new permanent service class element

Figure 6-25 Sample of JES2 command security table

WP6 general actions

In WP6, we covered all other actions that must be done that do not belong to any other work product. The following tasks were defined in the customer's environment:

- Create and education plan that is based on your environment
- Conduct basic JES2 education sessions all stakeholders
- Conduct more detailed education for infrastructure and operating teams
- Remove JES3 DLOG and set up OPERLOG, which is required by JES2
- Migrate any tools (commercial or custom-made) that still use JES3 DLOG

> **Attention:** In this WP, one of the key tasks that must be addressed is education. Experience shows that frequent education increases the acceptance of all stakeholders during the project.

The first basic education session must mandatory for all stakeholders in your company and conducted at the beginning of the project. Consider offering this educational session in different languages, if needed.

During the JES2 migration project, consider offering more detailed education sessions for expert teams, such as system engineering or system controlling (operations). The types of topics that must be addressed in detailed education sessions are listed in Table 6-23.

Table 6-23 Special education topics

Stakeholder	Education topics
JES2 system engineering	► JES2 start and shutdown in detail ► Important control block ► How to handle emergency situations: – SPOOL exhausted – CKPT damage – outage of an SPOOL volume ► Standard actions: – relocated CKPT – adding SPOOL – managing different start types – most important configuration settings

Stakeholder	Education topics
System controlling; for example, operations	► JES2 start and stop ► Explain default JES2 messages ► Explain useful JES2 commands to control the system ► JES2 start types

At the end of the project, we conducted a mandatory refresh session for all stakeholders that use JES2.

> **Tip:** For the success of the project, it might be necessary to request an acknowledgment from every department that is affected to ensure that all stakeholders are ready for the migration.

WP7 system automation

Almost all customers have a powerful system automation process in place to reduce manual intervention for known processing. In JES2, we managed by using automation for the relocation of the JES3 global processor during planned or unplanned outages of z/OS systems.

This subproject included the following tasks:

- Analyzed all JES3-related automatic actions.
- Define more JES2 surveillance actions (new JES2 messages).
- Removed no longer needed JES3 processing options.
- Adjusted all runbooks that are placed on the system control site.

The first task is to collect and document all JES3-related automation items that are defined in your environment. Then, you can categorize that list based on the following factors:

- No change: Can be used unchanged.
- Change: Must be adjusted according the JES2 (new message number).
- Delete: No longer needed with JES2.
- Add: New messages must be defined.

> **Tip:** For this task, it is helpful to have a JES2 expert onsite to discuss how certain JES3 settings are working under JES2. That is, what JES2 appears for a defined action that must be handled by the system automation.

WP9 JES2 system design and setup

This subproject covers all aspects of defining, sizing, and setting up JES2 in your environment.

> **Tip:** This action should be done as soon as possible so that all stakeholders can test any types of items to migrate under JES2.

The following tasks must be completed:

- Define JES2 system layout:
  - Define naming conventions for all needed new started tasks for JES2
  - Define data set naming for SPOOL and CKPT data sets or CF structures
  - Calculate SPOOL and CKPT sizes that are based on your JES3 sizes

- Create JES2 system environment according to your company defaults:
  - Allocate SPOOL space on dedicated DASD volumes
  - Allocate CKPT datasets or CF structures
  - Create STC JCL in your PROCLIBs
  - Create JES2 PARMLIBs
  - Define all of your NJE nodes and adapt the names from your JES3 node names
- Create a migration concept

Important: After finishing the JES2 setup, start any created JES2 instance with JES2 cold start to initialize your SPOOL. This process saves you time during the migration process. JES2 can be started as a secondary subsystem next to JES3 as the primary.

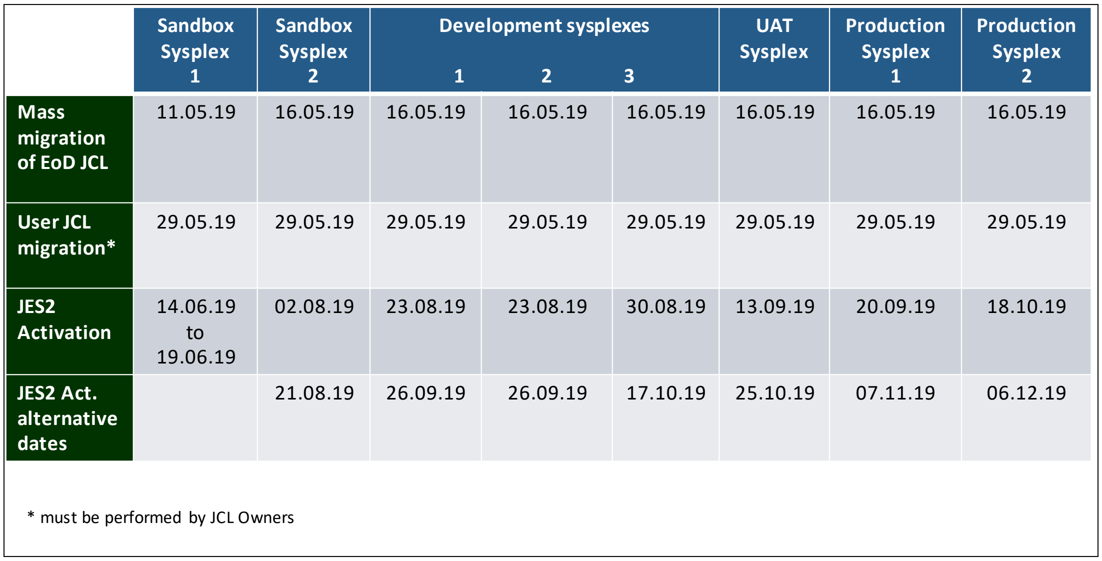
6.14.2 Project timeframe

In this section, we provide information about the time schedule in a customer's migration project.

Important: Start the project early in the year to avoid problems with end-of-year freeze.

The estimated time that you need strongly depends on your installation and the needed migration steps. In a customer's environment, much time was needed to re-create the program that was needed for managing printing.

Another important issue is how many human resources are available for the project. Because almost all of the migration steps can be done in parallel, the estimated time for the project depends on the available people power. An example of the time frames that are used in the case study is shown in Figure 6-26.



	Sandbox Sysplex 1	Sandbox Sysplex 2	Development sysplexes			UAT Sysplex	Production Sysplex 1	Production Sysplex 2
			1	2	3			
Mass migration of EoD JCL	11.05.19	16.05.19	16.05.19	16.05.19	16.05.19	16.05.19	16.05.19	16.05.19
User JCL migration*	29.05.19	29.05.19	29.05.19	29.05.19	29.05.19	29.05.19	29.05.19	29.05.19
JES2 Activation	14.06.19 to 19.06.19	02.08.19	23.08.19	23.08.19	30.08.19	13.09.19	20.09.19	18.10.19
JES2 Act. alternative dates		21.08.19	26.09.19	26.09.19	17.10.19	25.10.19	07.11.19	06.12.19

* must be performed by JCL Owners

Figure 6-26 Migration timeframe

The following timeframes are shown in Figure 6-26 on page 191:

JCL Mass Migration The checkpoint is part of the entire production JCL that often is managed by a professional BATCH management system. This checkpoint often is not complex because all JCL libraries are known.

User JCL Migration Before any JES2 migration, any stakeholder must finish their migration of JCL and JECL and report as such to the project team.

JES2 Activation This date is the primary migration date for the specific sysplex.

JES2 alternative date This date is the planned backup date if the primary date is not working for any reason.

Part 3

Appendixes

In Part 3, we provide some useful samples to help with migration and that can be used to explore new JES2 features after migration.

This part includes the following appendixes:

A

Sample JES3 exit to analyze JECL usage

This appendix includes sample code for JES3 user exit 33 that helps you detect the use of JCL or JECL statements that require reviewing and possibly replacing as part of the move to JES2.

Copyright license and permission to copy: This appendix contains a sample application program in source language that illustrates programming techniques. You might copy, modify, and distribute this sample program in any form without payment to IBM, for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample program is written. This example has not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of this program.

Sample JES3 user exit 33

Sample code for detecting JCL and JECL is shown in Example A-1.

Example A-1 Sample code for detecting JCL and JECL

```
UX33     TITLE 'JES3 CONTROL STATEMENT USER EXIT'                       00010000
IATUX33  AMODE 31                                                       00011000
IATUX33  RMODE ANY                                                      00012000
         IATYASM                                                        00013000
*START OF SPECIFICATIONS*********************************************** 00020000
*                                                                     * 00021000
* MODULE-NAME = IATUX33                                               * 00022000
*                                                                     * 00023000
* $MOD(IATUX33)    PROD(JES3):                                        * 00030000
*                                                                     * 00040000
* DESCRIPTIVE NAME=                                                   * 00050000
*           JES3 CONTROL STATEMENT USER EXIT                          * 00060000
*                                                                     * 00070000
* *01* PROPRIETARY STATEMENT=                                         * 00080000
*  **PROPRIETARY_STATEMENT********************************************* 00090000
*                                                                     * 00091000
*                                                                     * 00100000
*   LICENSED MATERIALS - PROPERTY OF IBM                              * 00110000
*   THIS MODULE IS "RESTRICTED MATERIALS OF IBM"                      * 00120000
*   5694-A01 COPYRIGHT IBM CORP. 2013                                 * 00130000
*                                                                     * 00140000
*   STATUS= HJS7780                                                   * 00150000
*                                                                     * 00160000
*  **END_OF_PROPRIETARY_STATEMENT************************************** 00170000
*                                                                     * 00171000
* Input Registers =                                                   * 00190000
*           R0    Irrelevant                                          * 00200000
*           R1    Address of current JCL statement                    * 00210000
*           R2-R9 Irrelevant                                          * 00220000
*           R10   IATUX33 base register                               * 00230000
*           R11   IATYFCT address                                     * 00240000
*           R12   IATYTVT address                                     * 00250000
*           R13   IATYISD input service data area                     * 00260000
*           R14   Return address                                      * 00270000
*           R15   Entry point address                                 * 00280000
*                                                                     * 00290000
* Entry purpose =                                                     * 00300000
*           IATUX33 is entered for each logical record of JCL         * 00310000
*           EXEC statements and for JES3 control statements           * 00320000
*           except DATASET/ENDDATASET.                                * 00330000
*                                                                     * 00340000
*                                                                     * 00350000
* Input =   R1 points to the current JCL record                       * 00360000
*                                                                     * 00370000
* Exit  =   ARETURN=0                                                 * 00380000
*                                                                     * 00390000
*           *** R1 MUST NOT BE CHANGED ***                            * 00400000
*                                                                     * 00410000
* Output =  JES3 JECL statements have been tracked.                   * 00420000
*                                                                     * 00430000
*                                                                     * 00440000
*END OF SPECIFICATIONS************************************************* 01350000
*                                                             *         01370000
*    Turn tracking on:     SETCON TRACKING=ON                 *         01373000
*                                                             *         01374000
*    Turn tracking off:    SETCON TRACKING=OFF                 *         01375000
*                                                             *         01376000
*    Display data:         DISPLAY OPDATA,TRACKING             *         01377000
*                                                             *         01378000
*    Clear tracked data by turning tracking off and then on.  *         01379000
*                                                             *         01380000
***************************************************************         01381000
         COPY  IATYGLOB                                                 01382000
IATUX33  START                                                          01383000
         TITLE 'JES3 General Equates'                                   01384000
         IATYEQU                                                        01385000
         TITLE 'JES3 General Registers'                                 01386000
         IATYREG                                                        01387000
         TITLE 'JES3 TVT'                                               01388000
         IATYTVT                                                        01389000
         TITLE 'JES3 Job Control Table'                                 01390000
         IATYJCT                                                        01391000
         TITLE 'JES3 Input Service Data Area'                           01392000
         IATYISD                                                        01393000
         TITLE 'Security Control Block'                                 01394000
         IATYSEC                                                        01395000
         TITLE 'Tracking Facility Request Parameters'                   01396000
         CNZTRPL ,                                                      01397000
         TITLE 'Job Data Accounting Block'                              01398000
         IATYJDA                                                        01399000
         TITLE 'Local DSECT for info field'                             01400000
U33CNZDE DSECT                                                          01401000
DESC1    DC    CL3''               Eyecatcher = J3:                     01402000
DESC2D   DC    CL3''               Input service day                    01403000
DESC2H   DC    CL2''               Input service hour                   01404000
DESC2M   DC    CL2''               Input service minute                 01405000
DESC3    DC    CL2''               JECL card detected and space         01406000
DESC4    DC    CL8''               Job name                             01407000
DESC5    DC    CL8''               User ID/POE                          01408000
         TITLE 'JES3 Control Statement User Exit 33'                    01550000
*-------------------------------------------------------------*         01551000
*        IATUX33 entry point                                  *         01552000
*-------------------------------------------------------------*         01553000
IATUX33  CSECT                                                          01556000
         LR    R10,R15             Establish module base                01560000
         USING IATUX33,R10         Establish using for module           01570000
         USING IATISDT,R13         Input service work area              01580000
         IATYMOD BR=YES            Module entry point ID                01590000
         SPACE 1                                                        01600000
*-------------------------------------------------------------*         01610000
*        Save statement address and zero the CNZ parm area.   *         01620000
*-------------------------------------------------------------*         01630000
         LR    R9,R1               Save JCL statement address           01640000
         LA    R8,UX33WA           Get work area address                01650000
         USING TRPL,R8             CNZTRKR parameter list               01660000
INFO     USING U33CNZDE,TRPL_Track_Info  CNZTRKR info field             01670000
         XC    TRPL(TRPL_LEN),TRPL  Clear the parm list                 01680000
*-------------------------------------------------------------*         01690000
*        Is current statement //*MAIN or /*MAIN               *         01700000
*-------------------------------------------------------------*         01710000
         CLC   T33TMAN,0(R9)       Check for //*MAIN                    01720000
         JE    UX33C005            If yes, handle it                    01730000
         CLC   1+T33TMAN(L'T33TMAN-1),0(R9)  Check for /*MAIN           01740000
         JNE   UX33C220            If not, continue checking            01750000
```

```
UX33C005 DS    0H                                                       01760000
         MVC   INFO.DESC3,T33TMAO  Indicate MAIN                        01770000
         J     UX33T800            Go track stmt                        01780000
*-------------------------------------------------------------*         01790000
*        Is current statement //*PROCESS                      *         01800000
*-------------------------------------------------------------*         01810000
UX33C220 DS    0H                                                       01820000
         CLC   T33TPRC,0(R9)       Check for //*PROCESS                 01830000
         JE    UX33C225            If yes, handle it                    01840000
         CLC   1+T33TPRC(L'T33TPRC-1),0(R9)  Check for /*PROCESS        01850000
         JNE   UX33C240            If not, continue checking            01860000
UX33C225 DS    0H                                                       01870000
         MVC   INFO.DESC3,T33TPRO  Indicate PROCESS                     01880000
         J     UX33T800            Go track stmt                        01890000
*-------------------------------------------------------------*         01900000
*        Is current statement //*ENDPROCESS                   *         01910000
*-------------------------------------------------------------*         01920000
UX33C240 DS    0H                                                       01930000
         CLC   T33TEPR,0(R9)       Check for //*ENDPROCESS              01940000
         JE    UX33C245            If yes, handle it                    01950000
         CLC   1+T33TEPR(L'T33TEPR-1),0(R9)  Check for /*ENDPROCESS     01960000
         JNE   UX33C260            If not, continue checking            01970000
UX33C245 DS    0H                                                       01980000
         MVC   INFO.DESC3,T33TEPO  Indicate ENDPROCESS                  01990000
         J     UX33T800            Go track stmt                        02000000
*-------------------------------------------------------------*         02010000
*        Is current statement //*FORMAT                       *         02020000
*-------------------------------------------------------------*         02030000
UX33C260 DS    0H                                                       02040000
         CLC   T33TFRM,0(R9)       Check for //*FORMAT                  02050000
         JE    UX33C265            If yes, handle it                    02060000
         CLC   1+T33TFRM(L'T33TFRM-1),0(R9)  Check for /*FORMAT         02070000
         JNE   UX33C280            If not, continue checking            02080000
UX33C265 DS    0H                                                       02090000
         MVC   INFO.DESC3,T33TFRO  Indicate FORMAT                      02100000
         J     UX33T800            Go track stmt                        02110000
*-------------------------------------------------------------*         02120000
*        Is current statement //*NET                          *         02130000
*-------------------------------------------------------------*         02140000
UX33C280 DS    0H                                                       02150000
         CLC   T33TNET,0(R9)       Check for //*NET                     02160000
         JNE   UX33C300            If not, continue checking            02170000
         MVC   INFO.DESC3,T33TNEO  Indicate NET                         02180000
         J     UX33T800            Go track stmt                        02190000
*-------------------------------------------------------------*         02200000
*        Is current statement //*NETACCT                      *         02210000
*-------------------------------------------------------------*         02220000
UX33C300 DS    0H                                                       02230000
         CLC   T33TNTA,0(R9)       Check for //*NETACCT                 02240000
         JNE   UX33C320            If not, continue checking            02250000
         MVC   INFO.DESC3,T33TNTO  Indicate NETACCT                     02260000
         J     UX33T800            Go track stmt                        02270000
*-------------------------------------------------------------*         02280000
*        Is current statement //*ROUTE                        *         02290000
*-------------------------------------------------------------*         02300000
UX33C320 DS    0H                                                       02310000
         CLC   T33TRTE,0(R9)       Check for //*ROUTE                   02320000
         JNE   UX33C340            If not, continue checking            02330000
         MVC   INFO.DESC3,T33TRTO  Indicate ROUTE                       02340000
         J     UX33T800            Go track stmt                        02350000
```

```
*-------------------------------------------------------------*         02360000
*        Is current statement //*OPERATOR                     *         02370000
*-------------------------------------------------------------*         02380000
UX33C340 DS    0H                                                       02390000
         CLC   T33TOPR,0(R9)       Check for //*OPERATOR                02400000
         JE    UX33C345            If yes, handle it                    02410000
         CLC   1+T33TOPR(L'T33TOPR-1),0(R9)  Check for /*OPERATOR       02420000
         JNE   UX33T900            If not, done checking                02430000
UX33C345 DS    0H                                                       02440000
         MVC   INFO.DESC3,T33TOPO  Indicate OPERATOR                    02450000
         J     UX33T800            Go track stmt                        02460000
*-------------------------------------------------------------*         02470000
*        Track JES3 JECL statement usage                      *         02480000
*-------------------------------------------------------------*         02490000
UX33T800 DS    0H                                                       02500000
         MVC   TRPL_ACRO,=CL4'TRPL'  Set parm list eye catcher          02510000
         MVI   TRPL_VERSION,TRPL_K_JBB7727  Set parm list version       02520000
         ST    R10,TRPL_VIOLATORS_ADDR  Set event address               02530000
         MVC   INFO.DESC1,=CL3'J3:'  Indicate JES3 event                02540000
*-------------------------------------------------------------*         02550000
*        Include day and time the job went through input      *         02560000
*        service.  The format is DDDHHMM where:               *         02570000
*             DDD = day of the year                           *         02580000
*             HH  = hour of day                               *         02590000
*             MM  = minutes                                   *         02600000
*        This uses DESC4 as a work area.                      *         02610000
*-------------------------------------------------------------*         02620000
         L     R7,JDABADDR         Get JDAB                             02630000
         USING JDABSTRT,R7         JDABSTRT                             02640000
         UNPK  INFO.DESC2D,IRDATON  Set day                             02650000
         L     R15,IRTIMON         Get hundredths of seconds           +02660000
                                     since midnight                     02670000
         XR    R14,R14             Clear for divide                     02680000
         D     R14,=F'360000'      Get hours                            02690000
         CVD   R15,INFO.DESC4      Convert to packed dec                02700000
         OI    INFO.DESC4+7,X'0F'  Turn on sign bits                    02710000
         UNPK  INFO.DESC2H,INFO.DESC4+5(3)  Make printable              02720000
         LR    R15,R14             Move remainder to R15                02730000
         XR    R14,R14             Clear for divide                     02740000
         D     R14,=F'6000'        Get number of minutes                02750000
         CVD   R15,INFO.DESC4      Convert to packed dec                02760000
         OI    INFO.DESC4+7,X'0F'  Turn sign bits on                    02770000
         UNPK  INFO.DESC2M,INFO.DESC4+5(3)  Make printable              02780000
*-------------------------------------------------------------*         02790000
*        Add job name and user ID                             *         02800000
*-------------------------------------------------------------*         02810000
         MVC   INFO.DESC4,JDABJNAM  Indicate job name                   02820000
         MVC   INFO.DESC5,ISTUSID  Indicate user ID                     02830000
         DROP  R7                  JDABSTRT                             02840000
*-------------------------------------------------------------*         02850000
*        Track the event.                                     *         02860000
*-------------------------------------------------------------*         02870000
         CNZTRKR (R8)              Track the event                      02880000
*-------------------------------------------------------------*         02890000
*        Replace user ID with port of entry (POE).            *         02900000
*-------------------------------------------------------------*         02910000
         MVC   INFO.DESC5,ISDPOE   Set port of origin                   02920000
*-------------------------------------------------------------*         02930000
*        Track a second time, now with POE.                   *         02940000
*-------------------------------------------------------------*         02950000
```

```
         CNZTRKR (R8)              Track the event                      02960000
         DROP  R8                  CNZTRKR parms                        02970000
*-------------------------------------------------------------*         02980000
*        Setup for return                                     *         02990000
*-------------------------------------------------------------*         03000000
UX33T900 DS    0H                                                       03010000
         LR    R1,R9               Restore JCL statement address        03020000
         LA    R15,0               Always use normal return             03030000
         ARETURN                   Return to caller                     03040000
*-------------------------------------------------------------*         03050000
*        IATUX33 module work area                             *         03060000
*-------------------------------------------------------------*         03070000
UX33WA   DC    CL(TRPL_LEN)''                                           03080000
*-------------------------------------------------------------*         03090000
*        IATUX33 module constants                             *         03100000
*-------------------------------------------------------------*         03110000
T33TRTE  DC    CL9'//*ROUTE '                                           03120000
T33TRTO  DC    CL2'R'                                                   03130000
T33TFRM  DC    CL10'//*FORMAT '                                         03140000
T33TFRO  DC    CL2'F'                                                   03150000
T33TPRC  DC    CL11'//*PROCESS '                                        03160000
T33TPRO  DC    CL2'P'                                                   03170000
T33TEPR  DC    CL14'//*ENDPROCESS '                                     03180000
T33TEPO  DC    CL2'E'                                                   03190000
T33TMAN  DC    CL8'//*MAIN '                                            03200000
T33TMAO  DC    CL2'M'                                                   03210000
T33TOPR  DC    CL12'//*OPERATOR '                                       03220000
T33TOPO  DC    CL2'O'                                                   03230000
T33TNTA  DC    CL11'//*NETACCT '                                        03240000
T33TNTO  DC    CL2'A'                                                   03250000
T33TNET  DC    CL7'//*NET '                                             03260000
T33TNEO  DC    CL2'N'                                                   03270000
*-------------------------------------------------------------*         03280000
*        IATUX33 epilog                                       *         03290000
*-------------------------------------------------------------*         03300000
         IATXPTCH LT               Expand literals                      03310000
APARNUM  DC    CL7'       '        APAR number                          99999997
PTFNUM   DC    CL7'&J3REL  '       PTF number                           99999998
         END   IATUX33                                                  99999999
```

B

Comparison of JES3 and JES2 commands

This appendix contains a reference to the differences in the commands that are provided by JES3 and JES2. This information can be used by a team that is considering migrating from JES3 to JES2.

Changes to OPERCMDS profiles also are referenced, where applicable. Although this information is not a complete list of all possible commands, it provides examples of each type of command.

List of commonly used JES3 and JES2 commands

The JES commands that the operators are most likely to use frequently are listed in Table B-1. The JES3 command and the JES2 equivalent also are listed. Also, if you use SAF to protect your operator commands, the OPERCMDS profile that protects the commands are listed in the table.

Table B-1 Commands and OPERCMDS profiles

Type of command	JES3	JES2	OPERCMDS profile

Shutdown	`*RETURN`
	`*DUMP`
		`$P JES2`
		`$P JES2,ABEND`
		`$P JES2,ABEND,FORCE`
			JES3.STOP.RETURN
			JES3.STOP.DUMP
			JES2.STOP.SYS

Printer devices	`*S PRT`	`$S PRTn`	JES3.START.DEV.dev JES2.START.DEV
Job queue	`*F Q H`	`$HA`	JES3.MODIFY.Q JES2.MODIFYHOLD.JOB

Initiator	`*F G main G inits`
	
	`*I G main G init`
		`$S INnn-nn`
		`$P INnn-nn`
		`$D INnn-nn`
			JES3.MODIFY.G
			JES2.START.INITIATOR
			JES2.STOP.INITIATOR

MVS	`*I D D=dddd`	`MVS D U,,,dddd,1`	JES3.DISPLAY.D MVS.DISPLAY.*
Device related	`*X CR,IN=RMT01RD1,K`	`$S R1.RD1`	JES3.CALL.dspname JES2.START.RMT
Remote console	`*I O`	`no equivalent`	JES3.DISPLAY.O
Remote printer	`*S RMT01PR1`	`$S R1.PR1`	JES3.START.name JES2.START.RMT

Spool related	`*I Q S`
	`*I J=`*jobname*
	`*I A`
		`$D Q`
		`$D'`*jobname*`'`
		`$D A`
			JES3.DISPLAY.Q
			JES3.DISPLAY.JOB
			JES3.DISPLAY.A
			JES2.DISPLAY.JOB

Restart	`*R J=nnnn`	`$E Jnnnn`	JES3.RESTART.name JES2.RESTART.BAT
Job modify	`*F J=nnnn,C`	`$C Jnnnn,P`	JES3.MODIFY.JOB JES2.CANCEL.BAT

Job output	`*I U J=nnnn`	`$L Jn`
		`$L Tn`
		`$L Sn`
			JES3.DISPLAY.U
			JES2.DISPLAY.BATOUT
			JES2.DISPLAY.TSUOUT
			JES2.DISPLAY.STCOUT

Reroute job	`*F U J=nnnn,ND=dest`	`$R ALL,J=nnnn,R`	JES3.MODIFY.U JES2.ROUTE.JOBOUT
SPOOL		`$S SPOOL`	JES2.START.SPOOL
Unknown commands			JES3.UNKNOWN JES2.UNKNOWN

C

Sample SMF84 Report program

This appendix includes sample code for a SMF84 report program that collects the SMF record 84 subtype 21 and generates two different reports based on user PARM.

Copyright license and permission to copy: This appendix contains a sample application program in source language that illustrates programming techniques. You might copy, modify, and distribute this sample program in any form without payment to IBM, for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample program is written. This example has not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of this program.

Source code of SMF84RPT program

The SMF84RPT program is controlled by PARM on the EXEC card. The user can select the report to be generated by using one of the following parameters:

- MEM: Indicates that the program generates a report with all memory that was used by JES2 for each interval.
- RSU: Indicates that the program generates a report with all resource usage by JES2 for each interval that is available on SMF records.

The input to program SMF84RPT is the SMF dump data set that is generated by IFASMFDx program.

The program writes output to two data sets: a SYSOUT with the report generated and a SYSPRINT with program messages.

An example of JCL statements that are required to run the SMF84RPT program and produce a report from JES2 SMF84 records is shown in Example C-1. This sample is showing an execution that uses the MEM option on EXEC PARM.

Example C-1 Sample JCL to run the SMF84RPT program to generate a MEM usage report

```
//SMF84JOB JOB (),'SMF84 MEM REPORT',CLASS=B,MSGCLASS=X,DSENQSHR=ALLOW,
//         MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID
//*
//STEP01   EXEC PGM=SMF84RPT,PARM='MEM'
//STEPLIB  DD   DSN=your-load-library,DISP=SHR
//SMFOUT   DD   SYSOUT=*
//SMFPRINT DD   SYSOUT=*
//SMFIN    DD   DISP=SHR,DSN=your-input-smfdump-dataset
```

An example of a Memory Usage report that is produced by SMF84RPT when the user selects the MEM option on EXEC PARM parameter is shown in Example C-2. The report is generated to all intervals that are collected from the input SMF data set that is provided to program.

Example C-2 Sample of MEM usage report generated by SMF84RPT program

```
1SMF-DATE   SMF-TIME Z/VERSION SYSID JES  MEM_NAME     MEM_REGION  MEM_USE     MEM_LOW     MEM_HIGH    MEM_AVG
 ---------- -------- --------- ----- ---- ------------ ----------- ----------- ----------- ----------- -----------
 2018/06/09 00:00:15 SP7.2.3   SC74  JES2 <16M USER        9.192MB   819.200KB   819.200KB   819.200KB   819.200KB
                                          <16M SYSTEM      9.192MB   409.600KB   409.600KB   409.600KB   409.600KB
                                          >16M USER        1.605GB   847.164MB   847.164MB   847.164MB   847.164MB
                                          >16M SYSTEM      1.605GB    12.632MB    12.632MB    12.632MB    12.632MB
                                          >2G PRIVATE     16.383TB   856.064MB   856.064MB   856.064MB   856.064MB
 SMF-DATE   SMF-TIME Z/VERSION SYSID JES  MEM_NAME     MEM_REGION  MEM_USE     MEM_LOW     MEM_HIGH    MEM_AVG
 ---------- -------- --------- ----- ---- ------------ ----------- ----------- ----------- ----------- -----------
 2018/06/09 01:00:15 SP7.2.3   SC74  JES2 <16M USER        9.192MB   819.200KB   819.200KB   819.200KB   819.200KB
                                          <16M SYSTEM      9.192MB   409.600KB   409.600KB   413.696KB   409.601KB
                                          >16M USER        1.605GB   847.164MB   847.164MB   847.164MB   847.164MB
                                          >16M SYSTEM      1.605GB    12.632MB    12.632MB    12.660MB    12.632MB
                                          >2G PRIVATE     16.383TB   856.064MB   856.064MB   856.064MB   856.064MB
 SMF-DATE   SMF-TIME Z/VERSION SYSID JES  MEM_NAME     MEM_REGION  MEM_USE     MEM_LOW     MEM_HIGH    MEM_AVG
 ---------- -------- --------- ----- ---- ------------ ----------- ----------- ----------- ----------- -----------
 2018/06/09 02:00:15 SP7.2.3   SC74  JES2 <16M USER        9.192MB   819.200KB   819.200KB   819.200KB   819.200KB
                                          <16M SYSTEM      9.192MB   409.600KB   409.600KB   409.600KB   409.600KB
                                          >16M USER        1.605GB   847.164MB   847.164MB   847.164MB   847.164MB
                                          >16M SYSTEM      1.605GB    12.632MB    12.632MB    12.632MB    12.632MB
                                          >2G PRIVATE     16.383TB   856.064MB   856.064MB   856.064MB   856.064MB
 SMF-DATE   SMF-TIME Z/VERSION SYSID JES  MEM_NAME     MEM_REGION  MEM_USE     MEM_LOW     MEM_HIGH    MEM_AVG
 ---------- -------- --------- ----- ---- ------------ ----------- ----------- ----------- ----------- -----------
 2018/06/09 03:00:15 SP7.2.3   SC74  JES2 <16M USER        9.192MB   819.200KB   819.200KB   819.200KB   819.200KB
                                          <16M SYSTEM      9.192MB   409.600KB   409.600KB   409.600KB   409.600KB
                                          >16M USER        1.605GB   847.164MB   847.164MB   847.164MB   847.164MB
                                          >16M SYSTEM      1.605GB    12.632MB    12.632MB    12.632MB    12.632MB
                                          >2G PRIVATE     16.383TB   856.064MB   856.064MB   856.064MB   856.064MB
```

```
SMF-DATE   SMF-TIME Z/VERSION SYSID JES  MEM_NAME     MEM_REGION  MEM_USE     MEM_LOW     MEM_HIGH    MEM_AVG
---------- -------- --------- ----- ---- ------------ ----------- ----------- ----------- ----------- -----------
2018/06/09 04:00:15 SP7.2.3   SC74  JES2 <16M USER        9.192MB   819.200KB   819.200KB   819.200KB   819.200KB
                                         <16M SYSTEM      9.192MB   409.600KB   409.600KB   409.600KB   409.600KB
                                         >16M USER        1.605GB   847.164MB   847.164MB   847.164MB   847.164MB
                                         >16M SYSTEM      1.605GB    12.632MB    12.632MB    12.632MB    12.632MB
                                         >2G PRIVATE     16.383TB   856.064MB   856.064MB   856.064MB   856.064MB
```

An example of a Resource Usage report that is generated by SMF84RPT with information about all resources that are used by JES2 on specific time interval is shown in Example C-3.

Example C-3 Sample of Resource usage by JES report generated by SMF84RPT program

```
1SMF-DATE   SMF-TIME Z/VERSION SYSID JES  RSU_NAME RSU_LIMIT RSU_INUSE RSU_LOW  RSU_HIGH RSU_WARN RSU_OVER RSU_AVG
 ---------- -------- --------- ----- ---- -------- --------- --------- -------- -------- -------- -------- --------
 2018/06/09 00:00:15 SP7.2.3   SC74  JES2 BERT          2100       378      376      378   80%           0      377
                                          BSCB             0         0        0        0    0%           0        0
                                          BUFX            79         0        0        0   80%           0        0
                                          CKVR            50         0        0        1   80%           0        0
                                          CMBS           204         1        1        1   80%           0        1
                                          CMDS          1000         0        0        0   80%           0        0
                                          ICES            33         0        0        0   80%           0        0
                                          JNUM          9999       619      617      619   80%           0      618
                                          JOES         10000      1410     1408     1410   80%           0     1410
                                          JQES          3000       619      617      619   80%           0      618
                                          LBUF            47         0        0        0   80%           0        0
                                          NHBS           100         0        0        0   80%           0        0
                                          SMFB            51         0        0        0   80%           0        0
                                          TBUF           104         0        0        0    0%           0        0
                                          TGS          40017      8944     8942     8944   80%           0     8943
                                          TTAB             3         0        0        0   80%           0        0
                                          VTMB             0         0        0        0    0%           0        0
                                          ZJC           1000        44       44       44   80%           0       44
 SMF-DATE   SMF-TIME Z/VERSION SYSID JES  RSU_NAME RSU_LIMIT RSU_INUSE RSU_LOW  RSU_HIGH RSU_WARN RSU_OVER RSU_AVG
 ---------- -------- --------- ----- ---- -------- --------- --------- -------- -------- -------- -------- --------
 2018/06/09 01:00:15 SP7.2.3   SC74  JES2 BERT          2100       378      376      378   80%           0      377
                                          BSCB             0         0        0        0    0%           0        0
                                          BUFX            79         0        0        0   80%           0        0
                                          CKVR            50         1        0        1   80%           0        0
                                          CMBS           204         1        1        1   80%           0        1
                                          CMDS          1000         0        0        0   80%           0        0
                                          ICES            33         0        0        0   80%           0        0
                                          JNUM          9999       621      619      621   80%           0      620
                                          JOES         10000      1412     1410     1412   80%           0     1412
                                          JQES          3000       621      619      621   80%           0      620
                                          LBUF            47         0        0        0   80%           0        0
                                          NHBS           100         0        0        0   80%           0        0
                                          SMFB            51         0        0        0   80%           0        0
                                          TBUF           104         0        0        0    0%           0        0
                                          TGS          40017      8946     8944     8946   80%           0     8945
                                          TTAB             3         0        0        0   80%           0        0
                                          VTMB             0         0        0        0    0%           0        0
                                          ZJC           1000        44       44       44   80%           0       44
 SMF-DATE   SMF-TIME Z/VERSION SYSID JES  RSU_NAME RSU_LIMIT RSU_INUSE RSU_LOW  RSU_HIGH RSU_WARN RSU_OVER RSU_AVG
 ---------- -------- --------- ----- ---- -------- --------- --------- -------- -------- -------- -------- --------
 2018/06/09 02:00:15 SP7.2.3   SC74  JES2 BERT          2100       378      376      378   80%           0      378
                                          BSCB             0         0        0        0    0%           0        0
                                          BUFX            79         0        0        0   80%           0        0
                                          CKVR            50         1        0        1   80%           0        0
                                          CMBS           204         1        1        1   80%           0        1
                                          CMDS          1000         0        0        0   80%           0        0
                                          ICES            33         0        0        0   80%           0        0
                                          JNUM          9999       623      621      623   80%           0      622
                                          JOES         10000      1414     1412     1414   80%           0     1413
                                          JQES          3000       623      621      623   80%           0      622
                                          LBUF            47         0        0        0   80%           0        0
                                          NHBS           100         0        0        0   80%           0        0
                                          SMFB            51         0        0        0   80%           0        0
                                          TBUF           104         0        0        0    0%           0        0
                                          TGS          40017      8950     8946     8950   80%           0     8948
                                          TTAB             3         0        0        0   80%           0        0
                                          VTMB             0         0        0        0    0%           0        0
                                          ZJC           1000        44       44       44   80%           0       44
```

Source code for the SMF84RPT program that is used to extract the SMF84 Subtype 21 records and create two different reports that are based on user selection on PARM parameter of EXEC JCL card is shown in Example C-4.

Example C-4 Source code of SMF84RPT program

```
SMF84RPT CSECT                                                          00010000
SMF84RPT RMODE 24                                                       00020000
SMF84RPT AMODE 31                                                       00030000
*/*******************************************************************/* 00031075
*/* THIS PROGRAM IS PART OF JES3 TO JES2 MIGRATION GUIDE REDBOOK    */* 00031175
*/*                                                                 */* 00031275
*/* THE MAIN FUNTION OF THIS PROGRAM IS EXTRACT THE JES2 SMF        */* 00031375
*/* RECORD AND GENERATE REPORTS BASED ON USER SELECTION             */* 00031475
*/*                                                                 */* 00031575
*/* THE USER SELECTION IS BASED ON PARM= JCL PARAMENTER             */* 00031675
*/*       MEM - SPECIFIES THE REPORT GENERATION TO MEMORY USAGE     */* 00031775
*/*       RSU - SPECIFIES REPORT GENERATION FOR RESOURCE USAGE      */* 00031875
*/*                                                                 */* 00031975
*/* THE REQUIRED DDNAMES ARE:                                       */* 00032075
*/*       SMFPRINT - OUTPUT FILE TO PROGRAM PROCESSING MESSAGES     */* 00032175
*/*       SMFIN    - INPUT SMF DATASET                              */* 00032275
*/*       SMFOUT   - OUTPUT FILE WITH REPORT DATA                   */* 00032375
*/*                  THIS DATASET HAVE THE LRECL DYNAMICALLY        */* 00032475
*/*                  GENERATED BY PROGRAM BASED ON REPORT SELECTION */* 00032575
*/*******************************************************************/* 00034075
         SAVE  (14,12)                                                  00040000
         LR    R12,R15                                                  00050067
         LR    R2,R1                                                    00060068
         USING SMF84RPT,R12                                             00080067
         STORAGE OBTAIN,LENGTH=WORKLEN,LOC=31                           00090000
         ST    R13,4(R1)                                                00100000
         ST    R1,8(R13)                                                00110000
         LR    R13,R1                                                   00120000
         USING WORKAREA,R13                                             00130000
         OPEN  (SMFPRINT,(OUTPUT))                                      00140068
*/*******************************************************************/* 00140168
*/* VALIDATE THE PARAMETER PASSED BY PARM EXEC - DEFAULT BOTH       */* 00140268
*/*******************************************************************/* 00140368
         L     R2,0(R2)                                                 00140468
         LTR   R2,R2                                                    00140568
         BZ    ERROR_NO_PARM                                            00140668
         LH    R3,0(R2)                                                 00140770
         CH    R3,HALF_3                                                00140870
         BNE   ERROR_WRONG_PARM                                         00140968
VALIDATE_KEYWORD_MEM EQU *                                              00141068
         CLC   KEYWORD_MEM,2(R2)                                        00141168
         BNE   VALIDATE_KEYWORD_RSU                                     00141268
         OI    FLAG_REPORT_TYPE,FLAG_REPORT_MEM                         00141368
         B     START_PROCESSING                                         00141468
VALIDATE_KEYWORD_RSU EQU *                                              00141568
         CLC   KEYWORD_RSU,2(R2)                                        00141668
         BNE   ERROR_INVALID_PARM                                       00141768
         OI    FLAG_REPORT_TYPE,FLAG_REPORT_RSU                         00141868
*/*******************************************************************/* 00141968
*/* OPEN THE SYSPRINT FILE AND CALL THE SUBROUTINE TO BUILD REPORT  */* 00142068
```

```
*/*******************************************************************/* 00142168
START_PROCESSING EQU *                                                  00142268
         PERFORM BUILD_REPORT_TABLE,R                                   00150037
         B     MAIN_PROCESS                                             00160001
*/*******************************************************************/* 01230000
*/* SUBROUTINE TO CREATE A TABLE WITH REPORT OPTIONS TO BE PRINTED  */* 01240000
*/*******************************************************************/* 01250000
BUILD_REPORT_TABLE EQU *                                                01260058
         LA    R1,REPORT_TABLE                                          01270000
         USING REPORT_ENTRY,R1                                          01280000
         LA    R15,SMFOUT_RECORD_START                                  01290000
         XC    RECORD_LENGTH,RECORD_LENGTH                              01300064
NEXT_TABLE_REPORT EQU *                                                 01310000
         CLC   0(L'END_TABLE,R1),END_TABLE                              01320000
         BE    END_TABLE_REPORT                                         01330000
*/*******************************************************************/* 01334265
*/* GET LENGTH OF FIELD AND KEYWORD TO OUTPUT RECORD                */* 01334365
*/*******************************************************************/* 01334465
         ST    R15,REPORT_FIELD_ADDRESS                                 01340000
         XR    R0,R0                                                    01350000
         XR    R14,R14                                                  01351064
         IC    R0,REPORT_FIELD_LENGTH                                   01360000
         IC    R14,REPORT_KEYWORD_LENGTH                                01361064
         AR    R15,R0                                                   01370000
         CR    R14,R0                                                   01370164
         BNL   ADD_RECORD_LENGTH                                        01370264
         LR    R14,R0                                                   01370364
*/*******************************************************************/* 01370865
*/* VALIDATE THE USABILITY OF FIELD ON OUTPUT RECORD                */* 01370965
*/*******************************************************************/* 01371065
ADD_RECORD_LENGTH EQU *                                                 01371165
         LA    R14,1(R14)                                               01371265
         CLI   REPORT_FIELD_INUSE,X'0F'                                 01371365
         BE    FIELD_TO_USE                                             01371465
         CLC   REPORT_FIELD_INUSE,FLAG_REPORT_TYPE                      01371565
         BE    FIELD_TO_USE                                             01371665
         XC    REPORT_FIELD_INUSE,REPORT_FIELD_INUSE                    01371765
         XR    R14,R14                                                  01371865
*/*******************************************************************/* 01371965
*/* ADD FIELD LENGTH TO OUTPUT RECORD LENGTH                        */* 01372065
*/*******************************************************************/* 01372165
FIELD_TO_USE EQU *                                                      01372265
         AH    R14,RECORD_LENGTH                                        01372364
         STH   R14,RECORD_LENGTH                                        01373064
         LA    R1,L'REPORT_TABLE_ENTRY(R1)                              01380000
         B     NEXT_TABLE_REPORT                                        01400000
END_TABLE_REPORT EQU *                                                  01410000
         BR    R10                                                      01420058
         DROP  R1                                                       01430000
*/*******************************************************************/* 01800000
*/* MAIN PROCEDURE TO PROGRAM PROCESSING                            */* 01810000
*/*******************************************************************/* 01820000
MAIN_PROCESS EQU *                                                      01830000
         PERFORM OPEN_FILES,R                                           01831037
NEXT_SMFIN_RECORD EQU *                                                 01832000
```

```
         PERFORM GET_SMFIN,R                                            01840037
*/*******************************************************************/* 01890005
*/* PROCESS THE SMF 84 RECORD SECTIONS                              */* 01900075
*/*******************************************************************/* 01901005
PROCESS_SUBTYPE EQU *                                                   01902005
         ZAP   SMFOUT_LINE,PACK_60                                      01903018
         PERFORM PROCESS_MEM_SECTION,R                                  01904037
         PERFORM PROCESS_RSU_SECTION,R                                  01905037
         B     NEXT_SMFIN_RECORD                                        01907005
*/*******************************************************************/* 01908168
*/* SUBROUTINE TO READ AND SELECT THE SMF RECORDS FROM INPUT FILE   */* 01908275
*/*******************************************************************/* 01908368
GET_SMFIN EQU  *                                                        01908468
         GET   SMFIN                                                    01908568
         LR    R2,R1                                                    01908668
         ST    R2,SMF_RECORD_ADDRESS                                    01908768
         L     R15,SMF_RECORD_COUNT                                     01908868
         LA    R15,1(R15)                                               01908968
         ST    R15,SMF_RECORD_COUNT                                     01909068
*/*******************************************************************/* 01909168
*/* VALIDATE THE SMF RECORD READ                                    */* 01909275
*/*******************************************************************/* 01909368
         USING SMF84HDR,R2                                              01909468
         CLC   SMF84RTY,SMF_RECORD_TYPE                                 01909573
         BNE   GET_SMFIN                                                01909668
         CLC   SMF84STY+1(1),SMF_RECORD_SUBTYPE                         01909774
         BNE   GET_SMFIN                                                01909868
SMF_RECORD_SELECTED EQU *                                               01909968
         L     R15,SMF_RECORD_SELECT                                    01910068
         LA    R15,1(R15)                                               01910168
         ST    R15,SMF_RECORD_SELECT                                    01910268
         BR    R10                                                      01910368
*/*******************************************************************/* 01910405
*/* PROCESS THE INFORMATION FROM SMF RECORD HEADER                  */* 01911075
*/*******************************************************************/* 01920005
INIT_SMFOUT_HEADER EQU *                                                01930005
         L     R2,SMF_RECORD_ADDRESS                                    01931014
         USING SMF84HDR,R2                                              01940005
         MVC   SMFOUT_SYSID,SMF84SID                                    01950005
         SMFTIME SMF84TME,SMFOUT_TIME                                   01960005
         SMFDATE SMF84DTE,SMFOUT_DATE                                   01970005
         MVC   SMFOUT_JES,=CL4'JES2'                                    01970109
         CLC   SMF84SBS,=AL2(SMF84HAS)                                  01970311
         BNE   NEXT_SMFIN_RECORD                                        01970409
         LR    R9,R2                                                    01970509
         A     R9,SMF84PRS                                              01970609
         USING SMF84PRO,R9                                              01970709
         MVC   SMFOUT_MVSVERS,R84MVSRL                                  01970810
         BR    R10                                                      01980005
         DROP  R2                                                       01990005
         DROP  R9                                                       02000009
*/*******************************************************************/* 02070000
*/* PROCESS THE INFORMATION FROM SMF 84 MEM SECTION                 */* 02080075
*/*******************************************************************/* 02090000
PROCESS_MEM_SECTION EQU *                                               02100001
```

```
         TM    FLAG_REPORT_TYPE,FLAG_REPORT_MEM                         02100159
         BNOR  R10                                                      02100259
         PERFORM INIT_SMFOUT_HEADER,R                                   02100337
         L     R2,SMF_RECORD_ADDRESS                                    02100414
         USING SMF84HDR,R2                                              02100513
         LR    R9,R2                                                    02100605
         A     R9,SMF84J1O                                              02100713
         USING SMF84JRU,R9                                              02101005
         ICM   R3,15,R84J2RMO                                           02110001
         LTR   R3,R3                                                    02120000
         BZR   R10                                                      02130000
         XR    R4,R4                                                    02131000
         ICM   R4,3,R84J2RML                                            02132001
         XR    R5,R5                                                    02140000
         ICM   R5,3,R84J2RMN                                            02150001
         LTR   R5,R5                                                    02151000
         BZR   R10                                                      02152000
         LA    R3,0(R3,R9)                                              02160005
         USING R84MEMJ2,R3                                              02170001
*/*******************************************************************/* 02180000
*/* EDIT AND PRINT THE INFORMATION FROM SMF 84 MEM SECTION         */* 02190075
*/*******************************************************************/* 02200000
NEXT_MEM_SECTION EQU *                                                  02210001
         MVC   SMFOUT_MEM_NAME,R84MEM_NAME                              02211101
         LG    R1,R84MEM_REGION                                         02211580
         PERFORM STORAGE_CALC,R                                         02211680
         MVC   SMFOUT_MEM_REGION_V,EDIT_MASK_DEC                        02212085
         ED    SMFOUT_MEM_REGION_V,DOUBLE+4                             02212184
         MVC   SMFOUT_MEM_REGION_U,0(R15)                               02212282
         LG    R1,R84MEM_USE                                            02212385
         PERFORM STORAGE_CALC,R                                         02212485
         MVC   SMFOUT_MEM_USE_V,EDIT_MASK_DEC                           02212685
         ED    SMFOUT_MEM_USE_V,DOUBLE+4                                02212785
         MVC   SMFOUT_MEM_USE_U,0(R15)                                  02212885
         LG    R1,R84MEM_LOW                                            02212985
         PERFORM STORAGE_CALC,R                                         02213085
         MVC   SMFOUT_MEM_LOW_V,EDIT_MASK_DEC                           02213285
         ED    SMFOUT_MEM_LOW_V,DOUBLE+4                                02213385
         MVC   SMFOUT_MEM_LOW_U,0(R15)                                  02213485
         LG    R1,R84MEM_HIGH                                           02213585
         PERFORM STORAGE_CALC,R                                         02213685
         MVC   SMFOUT_MEM_HIGH_V,EDIT_MASK_DEC                          02213885
         ED    SMFOUT_MEM_HIGH_V,DOUBLE+4                               02213985
         MVC   SMFOUT_MEM_HIGH_U,0(R15)                                 02214085
         LG    R1,R84MEM_AVERAGE                                        02214185
         PERFORM STORAGE_CALC,R                                         02214285
         MVC   SMFOUT_MEM_AVERAGE_V,EDIT_MASK_DEC                       02214485
         ED    SMFOUT_MEM_AVERAGE_V,DOUBLE+4                            02214585
         MVC   SMFOUT_MEM_AVERAGE_U,0(R15)                              02214685
         PERFORM PRINT_REPORT,R                                         02214745
         PERFORM CLEAR_SMFOUT_RECORD,R                                  02215047
*/*******************************************************************/* 02310000
*/* PROCESS THE NEXT TRIPLE FROM SMF 84 MEM SECTION                */* 02320075
*/*******************************************************************/* 02330000
GET_MEM_SECTION EQU *                                                   02340001
```

```
         AR    R3,R4                                                    02370000
         BCT   R5,NEXT_MEM_SECTION                                      02380001
         BR    R10                                                      02390000
         DROP  R2                                                       02410013
         DROP  R3                                                       02410113
         DROP  R9                                                       02411013
*/*******************************************************************/* 02420001
*/* SUBROUTINE TO REDUCE AMOUNT OF MEMORY ON REPORT                 */* 02430080
*/*******************************************************************/* 02440001
STORAGE_CALC EQU *                                                      02440180
         LA    R15,STORAGE_UNIT                                         02440280
         LA    R14,5                                                    02440380
         XGR   R0,R0                                                    02440490
NEXT_DIVIDE EQU *                                                       02440580
         CG    R1,DOUBLE_1024                                           02440680
         BL    END_DIVIDE                                               02440791
         XGR   R0,R0                                                    02440891
         DLG   R0,DOUBLE_1024                                           02440980
         LA    R15,2(R15)                                               02441080
         CVDG  R1,DOUBLE                                                02441180
         BCT   R14,NEXT_DIVIDE                                          02441282
END_DIVIDE EQU *                                                        02441390
         SLL   R1,10                                                    02441490
         OR    R1,R0                                                    02441590
         CVD   R1,DOUBLE                                                02441690
         BR    R10                                                      02441780
*/*******************************************************************/* 02441880
*/* PROCESS THE RSU SECTION FROM SMF84 RECORD                       */* 02442080
*/*******************************************************************/* 02443080
PROCESS_RSU_SECTION EQU *                                               02450001
         TM    FLAG_REPORT_TYPE,FLAG_REPORT_RSU                         02450159
         BNOR  R10                                                      02450259
         PERFORM INIT_SMFOUT_HEADER,R                                   02450337
         L     R2,SMF_RECORD_ADDRESS                                    02450415
         USING SMF84HDR,R2                                              02450515
         LR    R9,R2                                                    02450613
         A     R9,SMF84J1O                                              02450713
         USING SMF84JRU,R9                                              02460005
         ICM   R3,15,R84J2RRO                                           02470001
         LTR   R3,R3                                                    02480001
         BZR   R10                                                      02490001
         XR    R4,R4                                                    02500001
         ICM   R4,3,R84J2RRL                                            02510016
         XR    R5,R5                                                    02520001
         ICM   R5,3,R84J2RRN                                            02530016
         LTR   R5,R5                                                    02540001
         BZR   R10                                                      02550001
         LA    R3,0(R3,R9)                                              02560005
         USING R84RSUJ2,R3                                              02570001
*/*******************************************************************/* 02580001
*/* EDIT AND PRINT THE INFORMATION FROM SMF 84 RSU SECTION          */* 02590075
*/*******************************************************************/* 02600001
NEXT_RSU_SECTION EQU *                                                  02610001
         MVC   SMFOUT_RSU_NAME,R84RSU_NAME                              02620037
         L     R15,R84RSU_LIMIT                                         02620138
         CVD   R15,DOUBLE                                                02620238
         MVC   SMFOUT_RSU_LIMIT,EDIT_MASK                                02620338
         ED    SMFOUT_RSU_LIMIT,DOUBLE+4                                 02620441
         L     R15,R84RSU_INUSE                                          02620540
         CVD   R15,DOUBLE                                                02620640
         MVC   SMFOUT_RSU_INUSE,EDIT_MASK                                02620740
         ED    SMFOUT_RSU_INUSE,DOUBLE+4                                 02620841
         L     R15,R84RSU_LOW                                            02620941
         CVD   R15,DOUBLE                                                02621041
         MVC   SMFOUT_RSU_LOW,EDIT_MASK                                  02621141
         ED    SMFOUT_RSU_LOW,DOUBLE+4                                   02621241
         L     R15,R84RSU_HIGH                                           02621341
         CVD   R15,DOUBLE                                                02621441
         MVC   SMFOUT_RSU_HIGH,EDIT_MASK                                 02621541
         ED    SMFOUT_RSU_HIGH,DOUBLE+4                                  02621641
         LH    R15,R84RSU_WARN                                           02621742
         CVD   R15,DOUBLE                                                02621842
         MVC   SMFOUT_RSU_WARN,EDIT_MASK+4                               02621943
         MVI   SMFOUT_RSU_WARN,X'40'                                     02622042
         ED    SMFOUT_RSU_WARN,DOUBLE+6                                  02622253
         MVI   SMFOUT_RSU_WARN+4,C'%'                                    02622354
         L     R15,R84RSU_OVER                                           02622442
         CVD   R15,DOUBLE                                                02622542
         MVC   SMFOUT_RSU_OVER,EDIT_MASK                                 02622642
         ED    SMFOUT_RSU_OVER,DOUBLE+4                                  02622742
         L     R15,R84RSU_AVERAGE                                        02622842
         CVD   R15,DOUBLE                                                02622942
         MVC   SMFOUT_RSU_AVERAGE,EDIT_MASK                              02623042
         ED    SMFOUT_RSU_AVERAGE,DOUBLE+4                               02623142
         PERFORM PRINT_REPORT,R                                          02624042
         PERFORM CLEAR_SMFOUT_RECORD,R                                   02625047
*/******************************************************************/* 02630001
*/* GET THE NEXT FIELD AVAILABLE ON SMF 84 RSU SECTION              */* 02640075
*/******************************************************************/* 02650001
GET_RSU_SECTION EQU *                                                    02660001
         AR    R3,R4                                                     02670001
         BCT   R5,NEXT_RSU_SECTION                                       02680001
         BR    R10                                                       02690001
         DROP  R2                                                        02710013
         DROP  R3                                                        02710113
         DROP  R9                                                        02711013
*/******************************************************************/* 02720005
*/* SUBROUTINE TO OPEN THE INPUT AND OUTPUT FILES TO BE PROCESSED    */* 02730005
*/******************************************************************/* 02740005
OPEN_FILES EQU *                                                         02750005
         MVC   SMFOUT+82(2),RECORD_LENGTH                                02770005
         XC    WORK_FULL,WORK_FULL                                       02780005
         MVC   WORK_FULL+2(2),RECORD_LENGTH                              02790005
         GETMSG 5,SMFPRINT_RECORD,SMFMSG                                 02800005
         EDITMK WORK_FULL,SMFPRINT_LRECL                                 02810005
         PERFORM PUT_SMFPRINT,R                                          02820037
         OPEN  (SMFIN,(INPUT),SMFOUT,(OUTPUT))                           02830005
         XC    SMF_RECORD_COUNT,SMF_RECORD_COUNT                         02850005
         XC    SMF_RECORD_SELECT,SMF_RECORD_SELECT                       02860005
         BR    R10                                                       02870005
```

```
*/*******************************************************************/*
*/* SET THE RETURN CODE TO 8 AND END THE PROGRAM                    */*
*/*******************************************************************/*
RETURN_RC08 EQU *
         MVC   RETURN_CODE,FULL_8
         B     CLOSE_FILES
*/*******************************************************************/*
*/* CLOSE THE FILES AND END THE PROGRAM RETURNING TO CALLER         */*
*/*******************************************************************/*
END_SMFIN EQU  *
         GETMSG 6,SMFPRINT_RECORD,SMFMSG
         SMFEDIT SMF_RECORD_COUNT,SMFPRINT_RECTOT
         SMFEDIT SMF_RECORD_SELECT,SMFPRINT_RECSEL
         XR    R15,R15
         IC    R15,SMF_RECORD_TYPE
         ST    R15,WORK_FULL
         SMFEDIT WORK_FULL,SMFPRINT_RECTYPE
         PERFORM PUT_SMFPRINT,R
*/*******************************************************************/*
*/* CLOSE THE FILES AND RELEASE STORAGE AREAS ACQUIRED              */*
*/*******************************************************************/*
CLOSE_FILES EQU *
         CLOSE (SMFIN)
         CLOSE (SMFOUT)
         TM    FLAG_PROC,FLAG_REPORT+FLAG_CNTL
         BNO   RETURN_CALLER
         L     R1,REPORT_ADDRESS
         LH    R2,RECORD_LENGTH
         STORAGE RELEASE,LENGTH=(2),ADDR=(1)
*/*******************************************************************/*
*/* CLOSE THE SYSPRINT FILE AND RELEASE WORKAREA RETURNING TO CALLER*/*
*/*******************************************************************/*
RETURN_CALLER EQU *
         GETMSG 8,SMFPRINT_RECORD,SMFMSG
         EDITMK RETURN_CODE,SMFPRINT_RC
         PERFORM PUT_SMFPRINT,R
         CLOSE (SMFPRINT)
         LR    R1,R13
         L     R13,4(R13)
         STORAGE RELEASE,ADDR=(1),LENGTH=WORKLEN
         L     R15,RETURN_CODE
         L     R14,12(R13)
         LM    R0,R12,20(R13)
         BR    R14
*/*******************************************************************/*
*/* SUBROUTINE TO PRINT AN OUTPUT RECORD INTO SMFOUT FILE           */*
*/*******************************************************************/*
PRINT_REPORT EQU *
         TM    FLAG_PROC,FLAG_REPORT
         BO    START_DETAIL_REPORT
         LH    R8,RECORD_LENGTH
         STORAGE OBTAIN,LENGTH=(8),LOC=31
         ST    R1,REPORT_ADDRESS
         OI    FLAG_PROC,FLAG_REPORT
START_DETAIL_REPORT EQU *
```

```
         CP    SMFOUT_LINE,PACK_60                                      04090000
         BL    PUT_DETAIL_RECORD                                        04100000
         ZAP   SMFOUT_LINE,PACK_0                                       04100100
         PERFORM CLEAR_OUTPUT_RECORD,R                                  04110037
         LA    R2,REPORT_TABLE                                          04120006
         USING REPORT_ENTRY,R2                                          04130000
         L     R8,REPORT_ADDRESS                                        04140000
         PERFORM CREATE_TEXT_RECORD,R                                   04150037
         PERFORM PUT_SMFOUT_RECORD,R                                    04160037
         PERFORM CLEAR_OUTPUT_RECORD,R                                  04170037
         LA    R2,REPORT_TABLE                                          04180006
         L     R8,REPORT_ADDRESS                                        04190000
         PERFORM CREATE_LINE_RECORD,R                                   04200037
         PERFORM PUT_SMFOUT_RECORD,R                                    04210037
         ZAP   SMFOUT_LINE,PACK_2                                       04220000
PUT_DETAIL_RECORD EQU *                                                 04230000
         PERFORM CLEAR_OUTPUT_RECORD,R                                  04240037
         LA    R2,REPORT_TABLE                                          04250006
         L     R8,REPORT_ADDRESS                                        04260000
         PERFORM CREATE_DETAIL_RECORD,R                                 04270037
         PERFORM PUT_SMFOUT_RECORD,R                                    04280037
         AP    SMFOUT_LINE,PACK_1                                       04290000
         BR    R10                                                      04300000
*/*******************************************************************/* 04310000
*/* CREATE A RECORD WITH SEPARATOR CHARACTER TO BE PRINTED         */* 04320000
*/*******************************************************************/* 04330000
CREATE_LINE_RECORD EQU *                                                04340000
         CLC   0(L'END_TABLE,R2),END_TABLE                              04350000
         BER   R10                                                      04360000
         CLI   REPORT_FIELD_INUSE,X'00'                                 04361062
         BE    NEXT_LINE_FIELD                                          04362062
         XR    R15,R15                                                  04370000
         XR    R14,R14                                                  04380000
         IC    R14,REPORT_FIELD_LENGTH                                  04390000
         IC    R15,REPORT_KEYWORD_LENGTH                                04400000
         BCTR  R14,0                                                    04410000
         BCTR  R15,0                                                    04420000
         MVI   0(R8),C'-'                                               04440000
         LR    R1,R8                                                    04450000
         LA    R8,1(R8)                                                 04451000
         CR    R15,R14                                                  04460000
         BNL   MOVE_LINE_FIELD                                          04470000
         LR    R15,R14                                                  04480000
MOVE_LINE_FIELD EQU *                                                   04490000
         BCTR  R15,0                                                    04500000
         EX    R15,MOVE_REPORT_FIELD                                    04510000
         LA    R8,2(R15,R8)                                             04520000
NEXT_LINE_FIELD EQU *                                                   04521059
         LA    R2,L'REPORT_TABLE_ENTRY(R2)                              04530000
         B     CREATE_LINE_RECORD                                       04540000
*/*******************************************************************/* 04550000
*/* CREATE A RECORD WITH COLUMNS NAMES TO BE PRINTED               */* 04560000
*/*******************************************************************/* 04570000
CREATE_TEXT_RECORD EQU *                                                04580000
         CLC   0(L'END_TABLE,R2),END_TABLE                              04590000
         BER   R10                                                      04600000
         CLI   REPORT_FIELD_INUSE,X'00'                                 04601162
         BE    NEXT_TEXT_FIELD                                          04602062
         XR    R15,R15                                                  04610000
         XR    R14,R14                                                  04620000
         IC    R14,REPORT_FIELD_LENGTH                                  04630000
         IC    R15,REPORT_KEYWORD_LENGTH                                04640000
         BCTR  R14,0                                                    04650000
         BCTR  R15,0                                                    04660000
         LA    R1,REPORT_KEYWORD_DATA                                   04670000
         EX    R15,MOVE_REPORT_FIELD                                    04680000
         CR    R15,R14                                                  04690000
         BNL   MOVE_TEXT_FIELD                                          04700059
         LR    R15,R14                                                  04710000
MOVE_TEXT_FIELD EQU *                                                   04720059
         LA    R8,2(R15,R8)                                             04730000
NEXT_TEXT_FIELD EQU *                                                   04731059
         LA    R2,L'REPORT_TABLE_ENTRY(R2)                              04740000
         B     CREATE_TEXT_RECORD                                       04750000
*/*******************************************************************/* 04760000
*/* CREATE A RECORD WITH DETAILS REPORT DATA TO BE PRINTED          */* 04770000
*/*******************************************************************/* 04780000
CREATE_DETAIL_RECORD EQU *                                              04790000
         CLC   0(L'END_TABLE,R2),END_TABLE                              04800000
         BER   R10                                                      04810000
         CLI   REPORT_FIELD_INUSE,X'00'                                 04810162
         BE    BYPASS_DETAIL_FIELD                                      04812062
         XR    R15,R15                                                  04820000
         XR    R14,R14                                                  04830000
         IC    R15,REPORT_FIELD_LENGTH                                  04840000
         IC    R14,REPORT_KEYWORD_LENGTH                                04850000
         BCTR  R14,0                                                    04860000
         BCTR  R15,0                                                    04870000
         L     R1,REPORT_FIELD_ADDRESS                                  04880000
         EX    R15,MOVE_REPORT_FIELD                                    04890000
         TM    FLAG_PROC,FLAG_NOTITLE_KEYWORD                           04891000
         BO    NEXT_DETAIL_FIELD                                        04892000
         CR    R15,R14                                                  04900000
         BNL   NEXT_DETAIL_FIELD                                        04910000
         LR    R15,R14                                                  04920000
NEXT_DETAIL_FIELD EQU *                                                 04930000
         LA    R8,1(R15,R8)                                             04930100
         TM    FLAG_PROC,FLAG_BREAK_KEYWORD                             04931000
         BNO   STEP_DETAIL_FIELD                                        04932000
         MVC   0(1,R8),SEPARATOR_CHAR                                   04932100
STEP_DETAIL_FIELD EQU *                                                 04933000
         LA    R8,1(R8)                                                 04940000
BYPASS_DETAIL_FIELD EQU *                                               04941059
         LA    R2,L'REPORT_TABLE_ENTRY(R2)                              04950000
         B     CREATE_DETAIL_RECORD                                     04960000
MOVE_REPORT_FIELD MVC 0(0,R8),0(R1)                                     04970000
*/*******************************************************************/* 04990000
*/* CLEAR THE RECORD OUTPUT AREA TO BE PRINTED                      */* 05000000
*/*******************************************************************/* 05010000
CLEAR_OUTPUT_RECORD EQU *                                               05020000
```

```
         L     R6,REPORT_ADDRESS                                        05030000
         LA    R14,BLANK_CHAR                                           05040000
         LH    R7,RECORD_LENGTH                                         05050000
         XR    R15,R15                                                  05060000
         ICM   R15,8,0(R14)                                             05070000
         MVCL  R6,R14                                                   05080000
         BR    R10                                                      05090000
*/******************************************************************/* 05091047
*/* CLEAR THE SMFOUT RECORD USED TO PRINT THE REPORT               */* 05092047
*/******************************************************************/* 05093047
CLEAR_SMFOUT_RECORD EQU *                                               05094047
         LA    R6,SMFOUT_RECORD_START                                   05095047
         LA    R14,BLANK_CHAR                                           05096047
         L     R7,=A(SMFOUT_L)                                          05097048
         XR    R15,R15                                                  05098047
         ICM   R15,8,0(R14)                                             05099047
         MVCL  R6,R14                                                   05099147
         BR    R10                                                      05099247
*/******************************************************************/* 05100000
*/* PUT A RECORD LINE ON SMFOUT OUTPUT FILE                        */* 05110000
*/******************************************************************/* 05120000
PUT_SMFOUT_RECORD EQU *                                                 05130000
         L     R8,REPORT_ADDRESS                                        05140000
         PUT   SMFOUT,0(R8)                                             05150000
         BR    R10                                                      05160000
*/******************************************************************/* 05170000
*/* PRINT A MESSAGE OF PROGRAM PROCESSING                          */* 05180000
*/******************************************************************/* 05190000
PUT_SMFPRINT EQU *                                                      05200000
         PUT   SMFPRINT,SMFPRINT_RECORD                                 05210000
         BR    R10                                                      05220000
*/******************************************************************/* 05680000
*/* SEND A ERROR MESSAGE AND END THE PROGRAM WIRH RETURN CODE 8     */* 05690068
*/******************************************************************/* 05700000
ERROR_NO_PARM EQU *                                                     05710068
         GETMSG 1,SMFPRINT_RECORD,SMFMSG                                05720068
         PERFORM PUT_SMFPRINT,R                                         05730070
         B     RETURN_RC08                                              05740068
ERROR_WRONG_PARM EQU *                                                  05741068
         GETMSG 2,SMFPRINT_RECORD,SMFMSG                                05742068
         PERFORM PUT_SMFPRINT,R                                         05742170
         B     RETURN_RC08                                              05743068
ERROR_INVALID_PARM EQU *                                                05744068
         GETMSG 3,SMFPRINT_RECORD,SMFMSG                                05745068
         PERFORM PUT_SMFPRINT,R                                         05745170
         B     RETURN_RC08                                              05746068
*/******************************************************************/* 05750000
*/* DEFINE DATA CONSTANTS TO BE USED BY PROGRAM CHECKING           */* 05760000
*/******************************************************************/* 05770000
         LTORG                                                          05780000
FULL_8          DC F'8'                                                 05870000
HALF_3          DC H'3'                                                 05880068
                DS 0D                                                   05881081
DOUBLE_1024     DC X'0000000000000400'                                  05890079
PACK_0          DC PL1'0'                                               05920000
```

```
PACK_1          DC PL1'1'                                               05930000
PACK_2          DC PL1'2'                                               05940000
PACK_60         DC PL2'60'                                              05950000
PACK_255        DC PL2'255'                                             05951000
SMF_84_SUBTYPE  DC H'21'                                                05952001
BLANK_CHAR      DC C' '                                                 06030000
SEPARATOR_CHAR  DC C' '                                                 06040000
RETURN_CODE     DC F'0'                                                 06041000
END_TABLE       DC XL4'FFFFFFFF'                                        06160000
EDIT_MASK       DC X'4020202020202120'                                  06161053
EDIT_MASK_DEC   DC X'40202021204B202020'                                06162087
SMF_RECORD_TYPE DC X'54'                                                06170073
SMF_RECORD_SUBTYPE DC X'15'                                             06180073
KEYWORD_MEM     DC CL3'MEM'                                             06190068
KEYWORD_RSU     DC CL3'RSU'                                             06200068
STORAGE_UNIT    DC C' BKBMBGBTB'                                        06210081
*/*******************************************************************/* 06240000
*/* DEFINE REPORT HEADER AREA                                       */* 06250075
*/*******************************************************************/* 06260000
REPORT_TABLE DS 0F                                                      06270000
    DC A(15),A(0),AL1(L'SMFOUT_DATE),AL1(8),CL10'SMF-DATE'              06280062
    DC A(15),A(0),AL1(L'SMFOUT_TIME),AL1(8),CL10'SMF-TIME'              06290062
    DC A(15),A(0),AL1(L'SMFOUT_MVSVERS),AL1(9),CL10'Z/VERSION'          06300062
    DC A(15),A(0),AL1(L'SMFOUT_SYSID),AL1(5),CL10'SYSID'                06300162
    DC A(15),A(0),AL1(L'SMFOUT_JES),AL1(3),CL10'JES'                    06301062
    DC A(1),A(0),AL1(L'SMFOUT_MEM_NAME),AL1(8),CL10'MEM_NAME'           06310062
    DC A(1),A(0),AL1(L'SMFOUT_MEM_REGION),AL1(10),CL10'MEM_REGION'      06320062
    DC A(1),A(0),AL1(L'SMFOUT_MEM_USE),AL1(7),CL10'MEM_USE'             06330062
    DC A(1),A(0),AL1(L'SMFOUT_MEM_LOW),AL1(7),CL10'MEM_LOW'             06340062
    DC A(1),A(0),AL1(L'SMFOUT_MEM_HIGH),AL1(8),CL10'MEM_HIGH'           06350062
    DC A(1),A(0),AL1(L'SMFOUT_MEM_AVERAGE),AL1(7),CL10'MEM_AVG'         06360062
    DC A(2),A(0),AL1(L'SMFOUT_RSU_NAME),AL1(8),CL10'RSU_NAME'           06370062
    DC A(2),A(0),AL1(L'SMFOUT_RSU_LIMIT),AL1(9),CL10'RSU_LIMIT'         06371062
    DC A(2),A(0),AL1(L'SMFOUT_RSU_INUSE),AL1(9),CL10'RSU_INUSE'         06380062
    DC A(2),A(0),AL1(L'SMFOUT_RSU_LOW),AL1(7),CL10'RSU_LOW'             06400062
    DC A(2),A(0),AL1(L'SMFOUT_RSU_HIGH),AL1(8),CL10'RSU_HIGH'           06410062
    DC A(2),A(0),AL1(L'SMFOUT_RSU_WARN),AL1(8),CL10'RSU_WARN'           06420062
    DC A(2),A(0),AL1(L'SMFOUT_RSU_OVER),AL1(8),CL10'RSU_OVER'           06430062
    DC A(2),A(0),AL1(L'SMFOUT_RSU_AVERAGE),AL1(7),CL10'RSU_AVG'         06440062
    DC XL4'FFFFFFFF'                                                    06690059
*/*******************************************************************/* 06810000
*/* DEFINE DBS'S TO DATA SET PROCESSING                             */* 06820000
*/*******************************************************************/* 06830000
SMFIN    DCB   DDNAME=SMFIN,DSORG=PS,MACRF=GL,BFTEK=A,                 X06860000
               EODAD=END_SMFIN                                          06870000
SMFPRINT DCB   DDNAME=SMFPRINT,DSORG=PS,MACRF=PM,                      X06880000
               LRECL=133,RECFM=FBA,BLKSIZE=0                            06890000
SMFOUT DCB     DDNAME=SMFOUT,DSORG=PS,MACRF=PM,                        X06900000
               LRECL=0,RECFM=FB,BLKSIZE=0                               06910000
*/********************************************************************/ 06920093
*/* DEFINE AREA  WITH MESSAGE TO BE DISPLAIED ON PROGRAM            */ 06930093
*/********************************************************************/ 06931093
SMFMSG   DS    0F                                                       06932093
SMF001E  DC    CL133' SMF001E - PARM IS MISSING'                        06933093
SMF002E  DC    CL133' SMF002E - WRONG PARM PASSED'                      06934093
```

```
SMF003E  DC    CL133' SMF003E - INVALID PARM USED'                      06935093
SMF004I  DC    CL133' SMF004I - KEYWORD PARAMETER SELECTED WAS '        06936093
SMF005I  DC    CL133' SMF005I - REPORT WILL BE GENERATED WITH RECORD LEX06937093
               NGTH OF'                                                 06938093
SMF006I  DC    CL133' SMF006I - READ FROM SMF A TOTAL OF XXXXXXXXX RECOX06939093
               RDS AND PROCESSED XXXXXXXXX RECORDS TYPE'                06939193
SMF007E  DC    CL133' SMF007E - CONTROL CARD KEYWORD XXXXXXXXXX IS INVAX06939293
               LID'                                                     06939393
SMF008I  DC    CL133' SMF008I - PROGRAM ENDED WITH RETURN CODE'         06939493
*/*******************************************************************/* 06940000
*/* DEFINE DSECT WORKAREA TO VARIABLES USED BY PROGRAM              */* 06950000
*/*******************************************************************/* 06960000
WORKAREA    DSECT                                                       06970000
SAVEAREA       DS 18F                                                   06980000
DOUBLE         DS D                                                     06990000
               ORG DOUBLE                                               06991044
DOUBLE_WORK    DS XL16                                                  07000044
WORK_FULL      DS F                                                     07010000
REPORT_ADDRESS DS F                                                     07030000
RECORD_LENGTH  DS H                                                     07030164
SMF_RECORD_ADDRESS DS F                                                 07031014
SMF_RECORD_COUNT DS F                                                   07050000
SMF_RECORD_SELECT DS F                                                  07051000
SMF_DATE_START  DS F                                                    07090400
SMF_DATE_END    DS F                                                    07090600
SMFOUT_LINE    DS PL2                                                   07090700
FLAG_PROC             DS X                                              07091000
FLAG_REPORT          EQU X'01'                                          07100000
FLAG_CNTL            EQU X'02'                                          07110000
FLAG_NOTITLE_KEYWORD EQU X'40'                                          07115000
FLAG_BREAK_KEYWORD   EQU X'80'                                          07116000
FLAG_REPORT_TYPE  DS X                                                  07117062
FLAG_REPORT_HEAD EQU X'08'                                              07118062
FLAG_REPORT_MEM  EQU X'01'                                              07118162
FLAG_REPORT_RSU  EQU X'02'                                              07119062
            PERFORM GENERATE                                            07181037
*/*******************************************************************/* 07190000
*/* DEFINE OUTPUT RECORD AREA TO PRINT PROGRAM MESSAGES             */* 07200001
*/*******************************************************************/* 07210000
SMFPRINT_RECORD DS CL133                                                07220000
                ORG SMFPRINT_RECORD+18                                  07230000
SMFPRINT_SMF_TYPE DS CL3                                                07240000
                ORG SMFPRINT_RECORD+34                                  07241000
SMFPRINT_SMF_SUBTYPE DS CL3                                             07242000
                ORG SMFPRINT_RECORD+36                                  07242100
SMFPRINT_RECTOT DS CL9                                                  07242200
                ORG SMFPRINT_RECORD+68                                  07243000
SMFPRINT_RECSEL DS CL9                                                  07244000
                ORG SMFPRINT_RECORD+42                                  07250000
SMFPRINT_RC     DS CL3                                                  07260000
                ORG SMFPRINT_RECORD+44                                  07261000
SMFPRINT_DATE_ERROR DS CL7                                              07262000
                ORG SMFPRINT_RECORD+50                                  07270000
SMFPRINT_DATES  DS CL10                                                 07280000
                ORG SMFPRINT_RECORD+58                                  07280100
```

```
SMFPRINT_LRECL  DS CL5                                                  07280200
                ORG SMFPRINT_RECORD+62                                  07281000
SMFPRINT_CHAR_EXPECTED DS CL1                                           07282000
                ORG SMFPRINT_RECORD+69                                  07282100
SMFPRINT_DATEE  DS CL10                                                 07282200
                ORG SMFPRINT_RECORD+74                                  07283000
SMFPRINT_CHAR_FOUND DS CL1                                              07284000
                ORG SMFPRINT_RECORD+91                                  07285000
SMFPRINT_RECTYPE DS CL3                                                 07286000
                ORG                                                     07290000
*/*******************************************************************/* 07450000
*/* DEFINE OUTPUT RECORD AREA TO WRITE REPORTING DATA               */* 07460075
*/*******************************************************************/* 07470000
SMFOUT_RECORD_START EQU *                                               07480000
SMFOUT_DATE       DS CL10                                               07490000
SMFOUT_TIME        DS CL8                                               07500001
SMFOUT_MVSVERS     DS CL8                                               07510001
SMFOUT_SYSID       DS CL4                                               07510101
SMFOUT_JES         DS CL4                                               07511001
SMFOUT_MEM_NAME    DS CL12                                              07520001
SMFOUT_MEM_REGION  DS CL11                                              07522087
                   ORG SMFOUT_MEM_REGION                                07522180
SMFOUT_MEM_REGION_V DS CL9                                              07522287
SMFOUT_MEM_REGION_U DS CL2                                              07522380
SMFOUT_MEM_USE     DS CL11                                              07522487
                   ORG SMFOUT_MEM_USE                                   07522586
SMFOUT_MEM_USE_V    DS CL9                                              07522687
SMFOUT_MEM_USE_U    DS CL2                                              07522786
SMFOUT_MEM_LOW     DS CL11                                              07523087
                   ORG SMFOUT_MEM_LOW                                   07523186
SMFOUT_MEM_LOW_V    DS CL9                                              07523287
SMFOUT_MEM_LOW_U    DS CL2                                              07523386
SMFOUT_MEM_HIGH    DS CL11                                              07523487
                   ORG SMFOUT_MEM_HIGH                                  07523586
SMFOUT_MEM_HIGH_V   DS CL9                                              07523687
SMFOUT_MEM_HIGH_U   DS CL2                                              07523786
SMFOUT_MEM_AVERAGE DS CL11                                              07524087
                   ORG SMFOUT_MEM_AVERAGE                               07524186
SMFOUT_MEM_AVERAGE_V DS CL9                                             07524287
SMFOUT_MEM_AVERAGE_U DS CL2                                             07524386
SMFOUT_RSU_NAME    DS CL8                                               07525001
SMFOUT_RSU_LIMIT   DS CL8                                               07526001
SMFOUT_RSU_INUSE   DS CL8                                               07527001
SMFOUT_RSU_LOW     DS CL8                                               07528001
SMFOUT_RSU_HIGH    DS CL8                                               07529001
SMFOUT_RSU_WARN    DS CL5                                               07529152
SMFOUT_RSU_OVER    DS CL8                                               07529201
SMFOUT_RSU_AVERAGE DS CL8                                               07530092
SMFOUT_L          EQU *-SMFOUT_RECORD_START                             07540048
WORKLEN  EQU *-WORKAREA                                                 08320000
*/*******************************************************************/* 08330000
*/* DEFINE DSECT TO MAPPING THE KEYWORD REPORT ENTRY                */* 08340000
*/*******************************************************************/* 08350000
REPORT_ENTRY DSECT                                                      08360000
REPORT_TABLE_ENTRY DS XL20                                              08370059
```

```
             ORG REPORT_TABLE_ENTRY                                  08380000
REPORT_FIELD_AVAIL    DS A                                           08390059
             ORG REPORT_FIELD_AVAIL                                  08390159
                      DS XL3                                         08390462
REPORT_FIELD_INUSE    DS X                                           08390562
REPORT_FIELD_ADDRESS  DS A                                           08391059
REPORT_FIELD_LENGTH   DS AL1                                         08400000
REPORT_KEYWORD_LENGTH DS AL1                                         08410000
REPORT_KEYWORD_DATA   DS CL10                                        08420000
*/*****************************************************************/* 08590000
*/* DEFINE MACRO DSECT MAPPING TO MAP THE SMF 84 RECORD            */* 08600075
*/*****************************************************************/* 08610000
         IAZSMF84 SUBTYPE=21                                         08620001
         YREGS                                                       08680000
         END   SMF84RPT                                              08690004
```

An example of a PERFORM macro that is used by SMF84RPT program to run branches in program processing that uses GR10 is shown in Example C-5.

Example C-5 Sample of PERFORM macro

```
MACRO
&NAME    PERFORM &LABEL,&R
         GBLA  &PRFINDX
*
         AIF   ('&LABEL' EQ '').E1
         AIF   ('&LABEL' EQ 'GENERATE').DEFINE
         AIF   ('&R' EQ 'R').RENT
*
         AIF   (&SYSOPT_RENT).RENT
&NAME    ST    10,F&SYSNDX
         B     PERF&SYSNDX
F&SYSNDX DS    F
PERF&SYSNDX BAL   10,&LABEL
         L     10,F&SYSNDX
         AGO   .END
.RENT    ANOP
&PRFINDX SETA  &PRFINDX+1
&NAME    ST    10,F_P#&PRFINDX
         BAL   10,&LABEL
         L     10,F_P#&PRFINDX
         AGO   .END
.DEFINE  ANOP
         LCLA  &N
.LOOP    ANOP
&N       SETA  &N+1
F_P#&N   DS F
         AIF   (&N LT &PRFINDX).LOOP
         AGO   .END
.E1      MNOTE 8,'*** LABEL MISSING ***'
.END     MEND
```

An example of a GETMSG macro that is used by the SMF84RPT program to get messages from GETMSG CSECT and then places it on the SYSPRINT output data set is shown in Example C-6.

Example C-6   Sample of GETMSG macro

```
MACRO
&NOME    GETMSG &MSG,&AREA,&CSECT
         LCLA  &A
&NOME    LA    15,&MSG
         BCTR  15,0
         MH    15,=AL2(L'&AREA)
         AIF   ('&CSECT'(1,1) EQ '(').REGOK
         A     15,=A(&CSECT)
         AGO   .MOVE
.REGOK   ANOP
&REG     SETC  '&CSECT'(2,1)
&REGNO   SETA  &REG
         AR    15,&REGNO
.MOVE    ANOP
         MVC   &AREA+0(L'&AREA),0(15)
.EXIT    MEND
```

An example of the SMFDATE macro that is used by SMF84RPT program to convert date from SMF Julian format to edited European Gregorian format is shown in Example C-7.

Example C-7   Sample of SMFDATE macro

```
MACRO
&LABEL   SMFDATE &DATEI,&DATEO
         LCLC  &GVALU
&GVALU   SETC  'D'.'&SYSNDX'
         UNPK  DTI&SYSNDX+2(5),&DATEI+1(3)
         MVC   DTI&SYSNDX+0(2),A2&SYSNDX
         CLI   &DATEI,X'00'
         BNE   &GVALU.A
         MVC   DTI&SYSNDX+0(2),A1&SYSNDX
&GVALU.A EQU   *
         PACK  SBL&SYSNDX+0(8),DTI&SYSNDX+0(4)
         DP    SBL&SYSNDX+0(8),P4&SYSNDX      DIVIDE YEAR BY FOUR
         CLI   SBL&SYSNDX+7,X'0C'
         BNE   &GVALU.B
         MVI   M&SYSNDX+2,X'1D'               ADJUST FEBRUARY
&GVALU.B EQU   *
         PACK  SBL&SYSNDX+0(8),DTI&SYSNDX+4(3)
         CVB   0,SBL&SYSNDX                   JULIAN DAY
         LA    1,M&SYSNDX
         XR    15,15
&GVALU.C EQU   *
         LA    1,1(1)
         IC    15,0(1)
         CR    0,15
         BNH   &GVALU.D
         SR    0,15
         B     &GVALU.C
&GVALU.D EQU   *
```

```
         CVD   0,SBL&SYSNDX                  R0 GREGORIAN DAY
         MVC   &DATEO+0(10),DTE&SYSNDX
         UNPK  &DATEO+8(2),SBL&SYSNDX+6(2)
         OI    &DATEO+9,X'F0'
         LA    15,M&SYSNDX
         SR    1,15
         CVD   1,SBL&SYSNDX                  R4 GREGORIAN MONTH
         UNPK  &DATEO+5(2),SBL&SYSNDX+6(2)
         OI    &DATEO+6,X'F0'
         MVC   &DATEO+0(4),DTI&SYSNDX
         MVI   M&SYSNDX+2,X'1C'              ADJUST FEBRUARY MONTH
         B     &GVALU.E
SBL&SYSNDX DC  D'0'
DTI&SYSNDX DC  CL7' '
DTE&SYSNDX DC  C'AAAA/MM/DD'
A1&SYSNDX DC   C'19'
A2&SYSNDX DC   C'20'
P4&SYSNDX DC   P'4'
M&SYSNDX  DC   XL13'001F1C1F1E1F1E1F1F1E1F1E1F'
         DS    C
&GVALU.E EQU   *
         AGO   .E
.E       MEND
```

An example of the SMFTIME macro that is used by SMF84RPT program to convert time from SMF format to editable values is shown in Example C-8.

Example C-8   Sample of SMFTIME macro

```
MACRO
         SMFTIME &TIMEI,&TIMEO
         LCLC  &GVALU
&GVALU   SETC  'T'.'&SYSNDX'
         AIF   ('&TIMEI' EQ '').EI
         AIF   ('&TIMEO' EQ '').EO
&GVALU   EQU   *
         XR    R1,R1
         ICM   R1,15,&TIMEI
         XR    0,0
         D     0,F1&SYSNDX
         XR    0,0
         D     0,F3&SYSNDX
         CVD   1,DBL&SYSNDX
         MVC   &TIMEO+0(8),TME&SYSNDX
         UNPK  &TIMEO+0(2),DBL&SYSNDX+6(2)
         OI    &TIMEO+1,X'F0'
         LR    1,0
         XR    0,0
         D     0,F6&SYSNDX
         CVD   1,DBL&SYSNDX
         UNPK  &TIMEO+3(2),DBL&SYSNDX+6(2)
         OI    &TIMEO+4,X'F0'
         CVD   0,DBL&SYSNDX
         UNPK  &TIMEO+6(2),DBL&SYSNDX+6(2)
         OI    &TIMEO+7,X'F0'
         B     &GVALU.E
```

```
DBL&SYSNDX DC   D'0'
TME&SYSNDX DC   C'HH:MM:SS'
F1&SYSNDX DC   F'100'
F3&SYSNDX DC   F'3600'
F6&SYSNDX DC   F'60'
&GVALU.E  DS   0H
         AGO   .END
.EI      MNOTE 8,'SM001E *** TIME INPUT FIELD NOT SPECIFIED ***'
         AGO   .END
.EO      MNOTE 8,'SM001E *** TIME OUTPUT FIELD NOT SPECIFIED ***'
.END     MEND
```

An example of the SMFEDIT macro that is used by SMFRPT84 program to edit numbers that are used to count are shown in Example C-9.

Example C-9   SMFEDIT sample for editing data on program

```
MACRO
&LABEL   SMFEDIT &INPUT,&OUTPUT                                         00020004
         LCLC  &GVALU                                                   00030004
         LCLA  &LEN                                                     00040007
&GVALU   SETC  'E'.'&SYSNDX'                                            00050004
         XR    15,15                                                    00150007
         ICM   15,15,&INPUT                                             00280004
         CVD   15,DBL&SYSNDX                                            00300004
         MVC   EDT&SYSNDX,MSK&SYSNDX                                    00310004
         ED    EDT&SYSNDX,DBL&SYSNDX                                    00320004
         LA    15,EDT&SYSNDX+20-L'&OUTPUT                               00330004
         MVC   &OUTPUT,0(15)                                            00340004
         B     &GVALU.E                                                 00350004
EDT&SYSNDX DC  XL20'00'                                                 00360004
MSK&SYSNDX DC  XL20'402020204B2020204B2020204B2020204B202120'           00370004
DBL&SYSNDX DC  D'0'                                                     00380004
&GVALU.E EQU   *                                                        00390004
         AGO   .END                                                     00400006
.END     MEND                                                           00410006
```

An example of the EDITMK macro that is used to edit numbering for SMF report fields is shown in Example C-10.

Example C-10   EDITMK sample macro

```
MACRO
&LABEL   EDITMK &INPUT,&OUTPUT
         LCLC  &GVALU
&GVALU   SETC  'E'.'&SYSNDX'
         AIF   ('&OUTPUT'(1,1) NE '(').NOREG
         AIF   ('&OUTPUT'(3,1) NE ')').NOREG
&REG     SETC  '&OUTPUT'(2,1)
&REGNO   SETA  &REG
         AIF   (&REGNO LT 2).BADBASE
         AIF   (&REGNO GT 9).BADBASE
         LR    14,&REGNO
         AGO   .OKREG
.NOREG   ANOP
         LA    14,&OUTPUT
```

```
.OKREG   ANOP
         XR    15,15
         ICM   15,15,&INPUT
         CVD   15,DBL&SYSNDX
         LA    1,WRK&SYSNDX+19
         MVC   WRK&SYSNDX,MSK&SYSNDX
         EDMK  WRK&SYSNDX,DBL&SYSNDX
         LA    15,WRK&SYSNDX+19
         SR    15,1
         EX    15,MVC&SYSNDX
         LA    1,1(14,15)
         B     &GVALU.E
MSK&SYSNDX DC  X'402020204B2020204B2020204B2020204B202120'
WRK&SYSNDX DC  CL20' '
DBL&SYSNDX DC  D'0'
MVC&SYSNDX MVC 0(1,14),0(1)
&GVALU.E EQU   *
         AGO   .E
.BADBASE ANOP
         MNOTE 8,'*** ERROR ON THE REGISTER SPECIFICATION'
.E       MEND
```

D

DJC conversion and JEC examples

In this appendix, we show the results of a simple test we made to verify the way JES2 is handling the JES3 /*NET statements. We also provide a simple Job Execution Control (JEC) example that is performing the same management as JES3 NETs.

This chapter includes the following topics:

DJC conversion test results

JES2 supports the DJC //*NET statement and most of the parameters. The following parameters are not supported:

- DEVPOOL=
- DEVRELSE=
- RELSCHCT=

To enable DJC support, use the following command sequence:

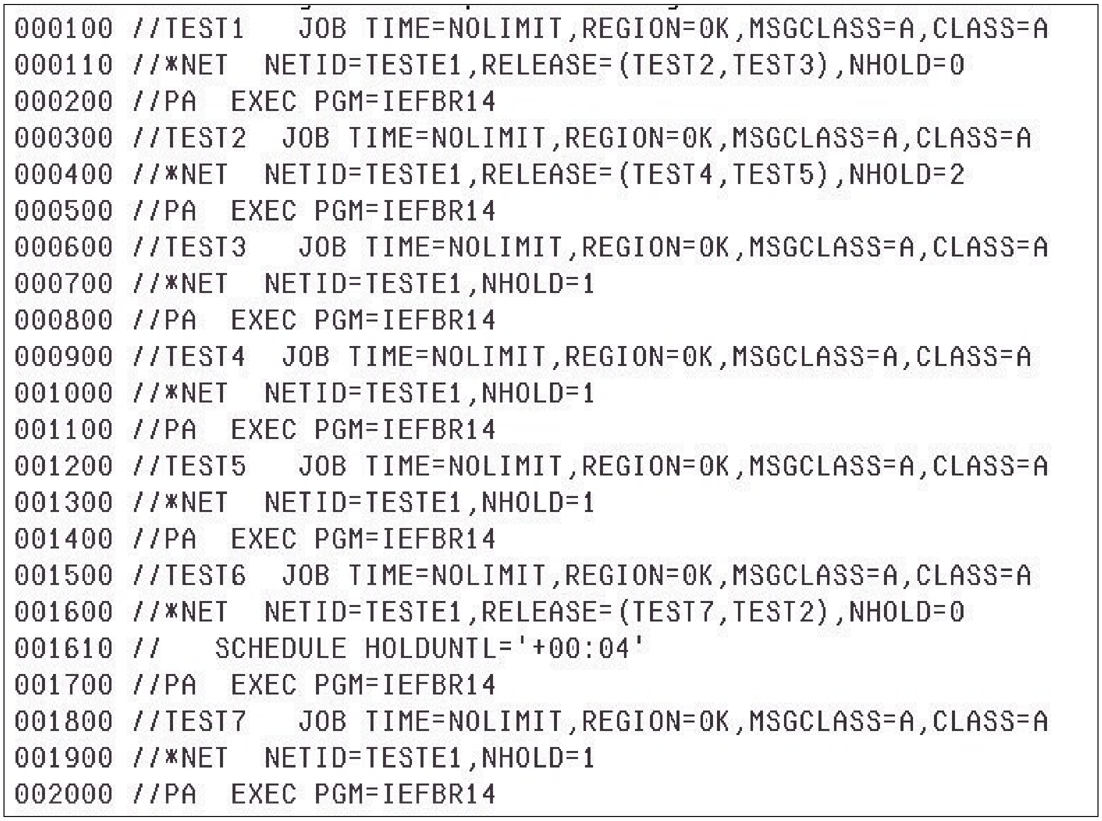
- `$T INPUTDEF,JES3JECL=PROCESS`
- `$T JECLDEF,JES3=(NET=PROCESS)`

When enabled, JES2 migrates JES3 //*NET JECL statements to the JES2 JEC job group support. A JEC job group is created to support a DJC semantics.

The JOBGROUP name is the NETID= value that is specified on the //*NET statement. The JOBGROUP that is created is marked as having a DJC statement origin. Marking it in this way allows JES2 to mimic the JES3 DJC runtime behavior. All job group commands can be used.

As with job groups, a logging job is created by using NETID. The logging job is a central place to collect messages that are related to important events in the life of the NETID and its constituent jobs. These events include jobs that are run or skipped and return codes.

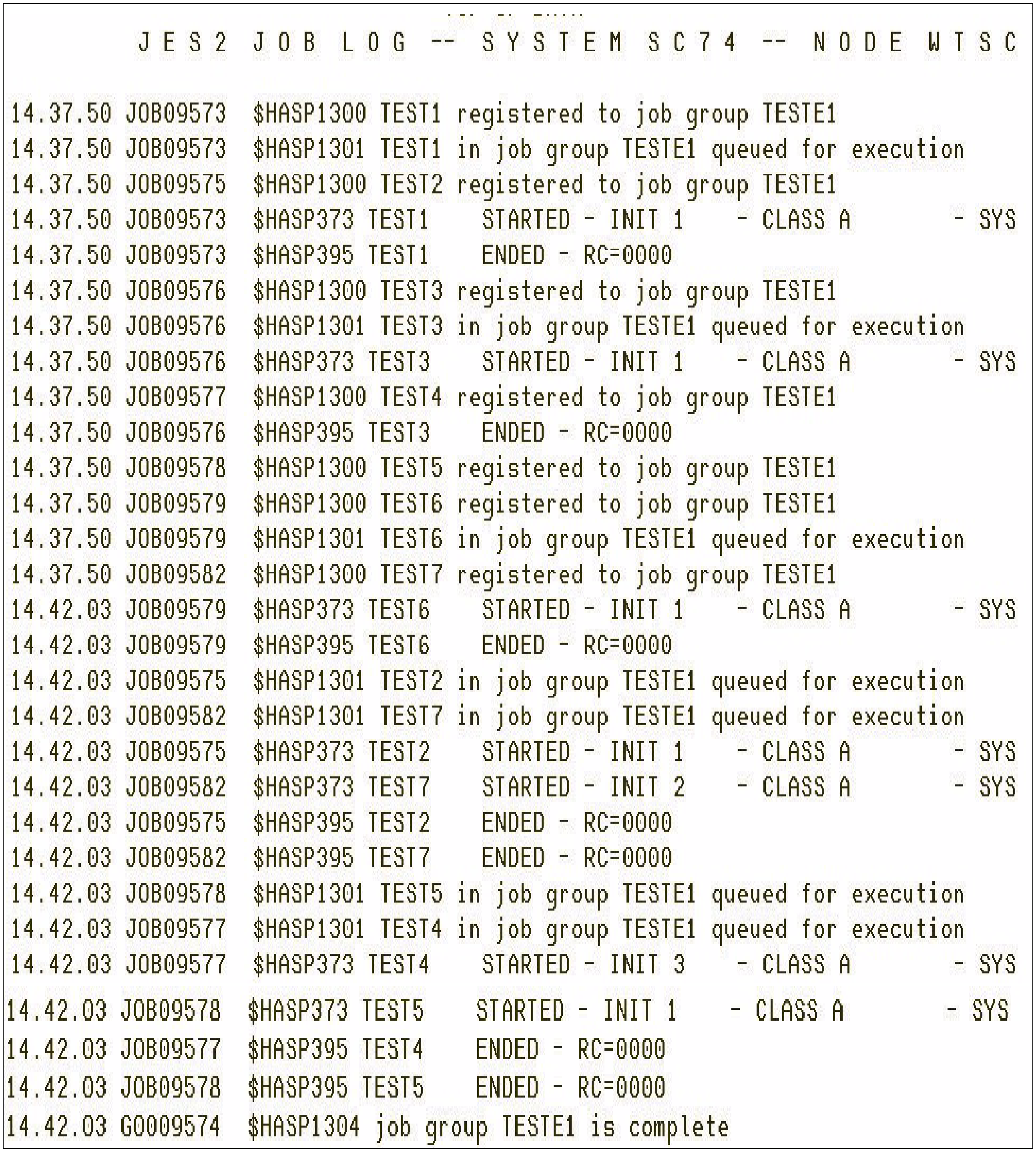
As an example, we created the simple job stream that is shown in Figure D-1. This job stream consists of seven jobs that belong to the same NETID. That is, their execution is interdependent. We also added the SCHEDULE JCL in Job TEST6 to observe the interaction between them.



```
000100 //TEST1   JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
000110 //*NET  NETID=TESTE1,RELEASE=(TEST2,TEST3),NHOLD=0
000200 //PA  EXEC PGM=IEFBR14
000300 //TEST2  JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
000400 //*NET  NETID=TESTE1,RELEASE=(TEST4,TEST5),NHOLD=2
000500 //PA  EXEC PGM=IEFBR14
000600 //TEST3   JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
000700 //*NET  NETID=TESTE1,NHOLD=1
000800 //PA  EXEC PGM=IEFBR14
000900 //TEST4  JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
001000 //*NET  NETID=TESTE1,NHOLD=1
001100 //PA  EXEC PGM=IEFBR14
001200 //TEST5   JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
001300 //*NET  NETID=TESTE1,NHOLD=1
001400 //PA  EXEC PGM=IEFBR14
001500 //TEST6  JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
001600 //*NET  NETID=TESTE1,RELEASE=(TEST7,TEST2),NHOLD=0
001610 //   SCHEDULE HOLDUNTL='+00:04'
001700 //PA  EXEC PGM=IEFBR14
001800 //TEST7   JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
001900 //*NET  NETID=TESTE1,NHOLD=1
002000 //PA  EXEC PGM=IEFBR14
```

Figure D-1   Simple job stream

The first thing you can see is that JES2 created a JOBGROUP with the same NETID name, as shown in Figure D-2.



```
        J E S 2  J O B  L O G  --  S Y S T E M  S C 7 4  --  N O D E  W T S C

14.37.50 JOB09573  $HASP1300 TEST1 registered to job group TESTE1
14.37.50 JOB09573  $HASP1301 TEST1 in job group TESTE1 queued for execution
14.37.50 JOB09575  $HASP1300 TEST2 registered to job group TESTE1
14.37.50 JOB09573  $HASP373 TEST1    STARTED - INIT 1    - CLASS A       - SYS
14.37.50 JOB09573  $HASP395 TEST1    ENDED - RC=0000
14.37.50 JOB09576  $HASP1300 TEST3 registered to job group TESTE1
14.37.50 JOB09576  $HASP1301 TEST3 in job group TESTE1 queued for execution
14.37.50 JOB09576  $HASP373 TEST3    STARTED - INIT 1    - CLASS A       - SYS
14.37.50 JOB09577  $HASP1300 TEST4 registered to job group TESTE1
14.37.50 JOB09576  $HASP395 TEST3    ENDED - RC=0000
14.37.50 JOB09578  $HASP1300 TEST5 registered to job group TESTE1
14.37.50 JOB09579  $HASP1300 TEST6 registered to job group TESTE1
14.37.50 JOB09579  $HASP1301 TEST6 in job group TESTE1 queued for execution
14.37.50 JOB09582  $HASP1300 TEST7 registered to job group TESTE1
14.42.03 JOB09579  $HASP373 TEST6    STARTED - INIT 1    - CLASS A       - SYS
14.42.03 JOB09579  $HASP395 TEST6    ENDED - RC=0000
14.42.03 JOB09575  $HASP1301 TEST2 in job group TESTE1 queued for execution
14.42.03 JOB09582  $HASP1301 TEST7 in job group TESTE1 queued for execution
14.42.03 JOB09575  $HASP373 TEST2    STARTED - INIT 1    - CLASS A       - SYS
14.42.03 JOB09582  $HASP373 TEST7    STARTED - INIT 2    - CLASS A       - SYS
14.42.03 JOB09575  $HASP395 TEST2    ENDED - RC=0000
14.42.03 JOB09582  $HASP395 TEST7    ENDED - RC=0000
14.42.03 JOB09578  $HASP1301 TEST5 in job group TESTE1 queued for execution
14.42.03 JOB09577  $HASP1301 TEST4 in job group TESTE1 queued for execution
14.42.03 JOB09577  $HASP373 TEST4    STARTED - INIT 3    - CLASS A       - SYS
14.42.03 JOB09578  $HASP373 TEST5    STARTED - INIT 1    - CLASS A       - SYS
14.42.03 JOB09577  $HASP395 TEST4    ENDED - RC=0000
14.42.03 JOB09578  $HASP395 TEST5    ENDED - RC=0000
14.42.03 G0009574  $HASP1304 job group TESTE1 is complete
```

Figure D-2 OUTPUT form TESTE1 JOBGROUP

All jobs in this NET were registered to TESTE1 JOBGROUP.

If you review the JECL statements, you see that TEST1 was submitted for immediate execution (the NHOLD value is zero). TEST2 depends on the execution of TEST1 and TEST6. TEST3 has only one dependency; it waits for TEST1 to finish.

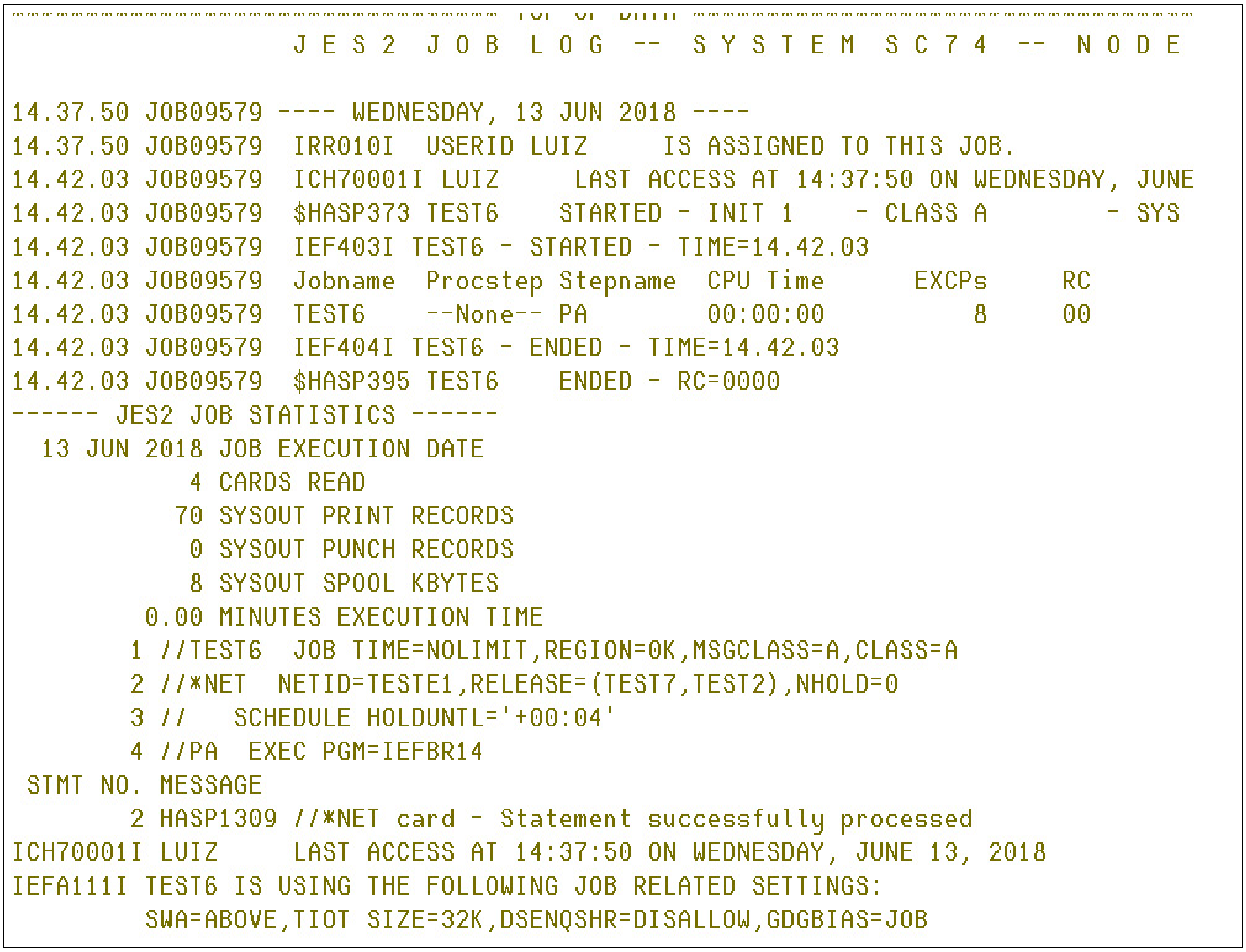
TEST4 also has only one dependency; it is released when TEST4 finishes. The same is true for TEST5.

Although TEST6 execution has no dependencies, it is delayed for four minutes because of the SCHEDULE JCL card. When TEST6 finishes, it releases jobs TEST7 and TEST2. TEST7 has only one dependency: the conclusion of TEST6.

As shown in Figure D-2 on page 228, TEST1 registered to group TESTE1 at 14:37:50. All other jobs registered to group TESTE1 at the same time (they were all submitted at the same time).

TEST1 began execution immediately; NHOLD=0. The other JOB that had NHOLD=0 was TEST6. However, the log indicates that it did not began its execution immediately; instead, it waited until 14:42:03 to be executed. The HOLDUNTIL parameter of the SCHEDULE JCL card delayed its execution for at least 4 minutes (see Figure D-3). As you can see, it is possible to add SCHEDULE JCL to a JES3 NET.

You also see that TEST3 began execution when TEST1 finished. The only dependency was the completion of TEST1.



```
********************************* TOP OF DATA **********************************
                   J E S 2  J O B  L O G  --  S Y S T E M  S C 7 4  --  N O D E

14.37.50 JOB09579 ---- WEDNESDAY, 13 JUN 2018 ----
14.37.50 JOB09579  IRR010I  USERID LUIZ     IS ASSIGNED TO THIS JOB.
14.42.03 JOB09579  ICH70001I LUIZ     LAST ACCESS AT 14:37:50 ON WEDNESDAY, JUNE
14.42.03 JOB09579  $HASP373 TEST6    STARTED - INIT 1    - CLASS A        - SYS
14.42.03 JOB09579  IEF403I TEST6 - STARTED - TIME=14.42.03
14.42.03 JOB09579  Jobname  Procstep Stepname  CPU Time      EXCPs     RC
14.42.03 JOB09579  TEST6    --None-- PA        00:00:00          8     00
14.42.03 JOB09579  IEF404I TEST6 - ENDED - TIME=14.42.03
14.42.03 JOB09579  $HASP395 TEST6    ENDED - RC=0000
------ JES2 JOB STATISTICS ------
  13 JUN 2018 JOB EXECUTION DATE
            4 CARDS READ
           70 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
            8 SYSOUT SPOOL KBYTES
         0.00 MINUTES EXECUTION TIME
        1 //TEST6  JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
        2 //*NET  NETID=TESTE1,RELEASE=(TEST7,TEST2),NHOLD=0
        3 //   SCHEDULE HOLDUNTL='+00:04'
        4 //PA  EXEC PGM=IEFBR14
 STMT NO. MESSAGE
        2 HASP1309 //*NET card - Statement successfully processed
ICH70001I LUIZ     LAST ACCESS AT 14:37:50 ON WEDNESDAY, JUNE 13, 2018
IEFA111I TEST6 IS USING THE FOLLOWING JOB RELATED SETTINGS:
         SWA=ABOVE,TIOT SIZE=32K,DSENQSHR=DISALLOW,GDGBIAS=JOB
```

Figure D-3   SYSOUT OF TEST6: Using JES2 JEC with JES3 NET

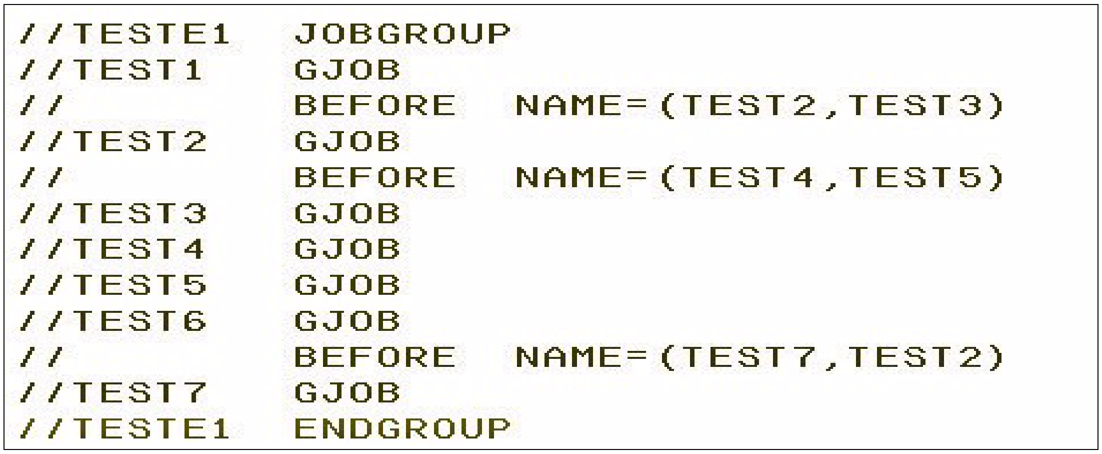
You can also see that TEST2 began its execution when TEST6 finished; the same process occurred with TEST7.

In Figure D-3 on page 229, you see the output of TEST4. That output shows a JES2 message HASP1309 indicates that the NET statement was successfully processed.

Using JES2 JEC

If you want to convert the job stream that is shown in Figure D-1 on page 227, you must create a JOB (see Figure D-4).

First, the job name is the same as the NETID. Then, you must to define by way of a GJOB statement all jobs that belong to this group.



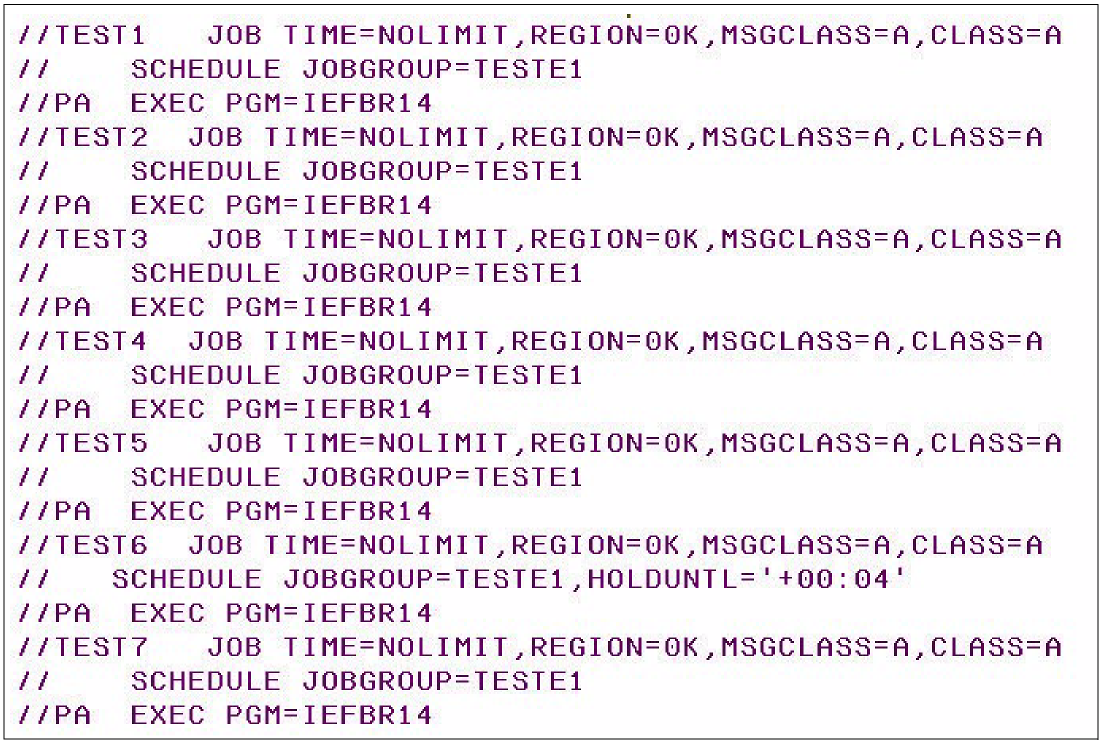
```
//TESTE1   JOBGROUP
//TEST1    GJOB
//         BEFORE   NAME=(TEST2,TEST3)
//TEST2    GJOB
//         BEFORE   NAME=(TEST4,TEST5)
//TEST3    GJOB
//TEST4    GJOB
//TEST5    GJOB
//TEST6    GJOB
//         BEFORE   NAME=(TEST7,TEST2)
//TEST7    GJOB
//TESTE1   ENDGROUP
```

Figure D-4 Example of a JOBGROUP job

For each job, you must define the relationship with other jobs; for example, TEST1 must execute before TEST2 and TEST3. TEST2 must execute before TEST4 and TEST5.

You must submit this job before submitting the jobs in the group for the dependency to take place.

After you submit your JOBGROUP, you can submit your job stream, as shown in Figure D-5.

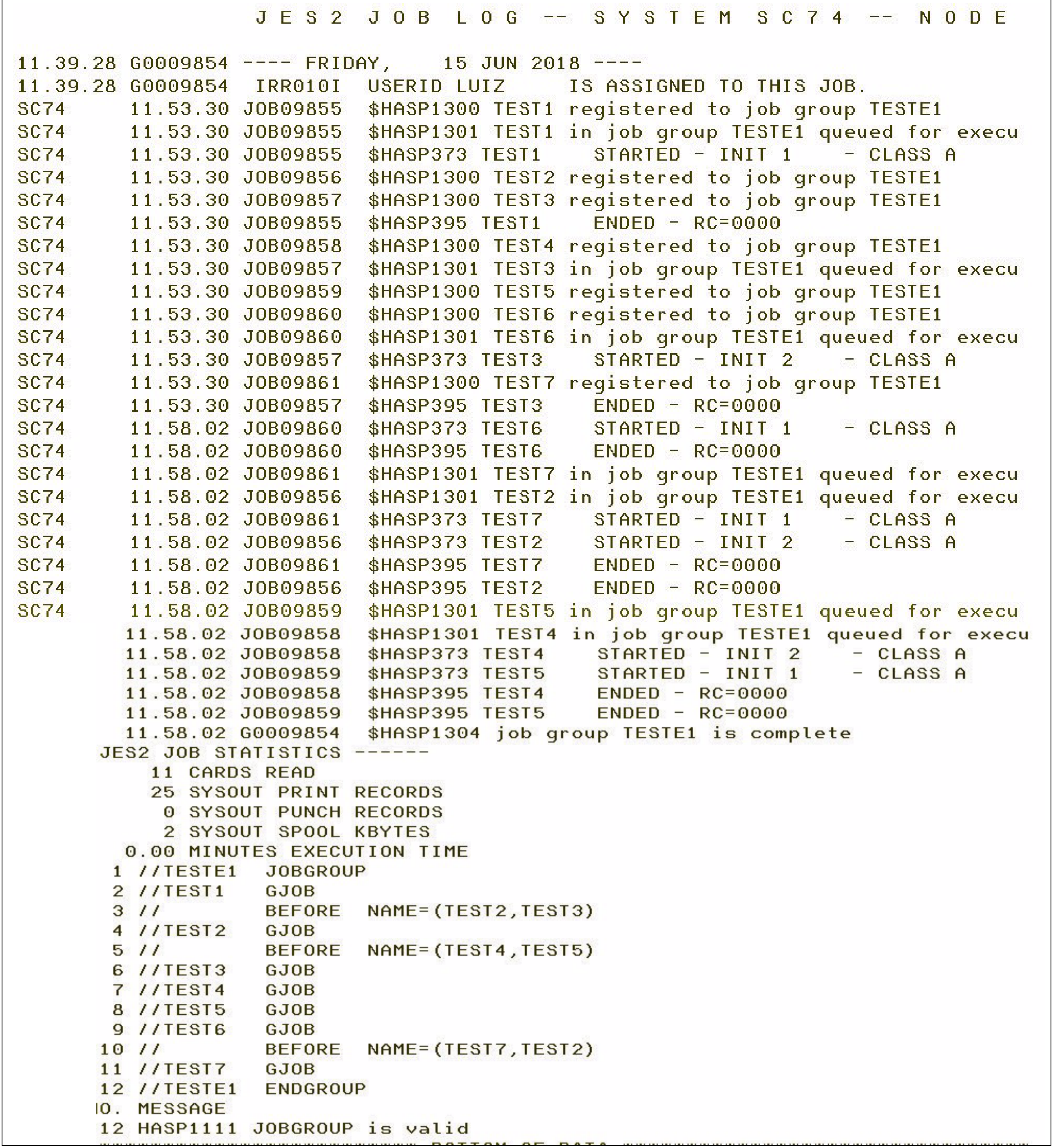


```
//TEST1   JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
//    SCHEDULE JOBGROUP=TESTE1
//PA  EXEC PGM=IEFBR14
//TEST2  JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
//    SCHEDULE JOBGROUP=TESTE1
//PA  EXEC PGM=IEFBR14
//TEST3   JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
//    SCHEDULE JOBGROUP=TESTE1
//PA  EXEC PGM=IEFBR14
//TEST4  JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
//    SCHEDULE JOBGROUP=TESTE1
//PA  EXEC PGM=IEFBR14
//TEST5   JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
//    SCHEDULE JOBGROUP=TESTE1
//PA  EXEC PGM=IEFBR14
//TEST6  JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
//   SCHEDULE JOBGROUP=TESTE1,HOLDUNTL='+00:04'
//PA  EXEC PGM=IEFBR14
//TEST7   JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A
//    SCHEDULE JOBGROUP=TESTE1
//PA  EXEC PGM=IEFBR14
```

Figure D-5   Jobstream with JES2 JCL statements

You must add a SCHEDULE JCL card with the JOBGROUP parameter for every job to associate the JOB to the corresponding JOBGROUP. You can also use other parameters in the SCHEDULER JCL statement, such as in TEST6.

In Example D-6, you see the messages that are generated for the execution of the jobs in group TESTE1. All jobs registered to job group TESTE1 at same time: 11:53:30. All were submitted together.



```
                  J E S 2   J O B   L O G  --  S Y S T E M   S C 7 4  --  N O D E

11.39.28 G0009854 ---- FRIDAY,    15 JUN 2018 ----
11.39.28 G0009854  IRR010I  USERID LUIZ     IS ASSIGNED TO THIS JOB.
SC74     11.53.30 JOB09855  $HASP1300 TEST1 registered to job group TESTE1
SC74     11.53.30 JOB09855  $HASP1301 TEST1 in job group TESTE1 queued for execu
SC74     11.53.30 JOB09855  $HASP373 TEST1    STARTED - INIT 1    - CLASS A
SC74     11.53.30 JOB09856  $HASP1300 TEST2 registered to job group TESTE1
SC74     11.53.30 JOB09857  $HASP1300 TEST3 registered to job group TESTE1
SC74     11.53.30 JOB09855  $HASP395 TEST1    ENDED - RC=0000
SC74     11.53.30 JOB09858  $HASP1300 TEST4 registered to job group TESTE1
SC74     11.53.30 JOB09857  $HASP1301 TEST3 in job group TESTE1 queued for execu
SC74     11.53.30 JOB09859  $HASP1300 TEST5 registered to job group TESTE1
SC74     11.53.30 JOB09860  $HASP1300 TEST6 registered to job group TESTE1
SC74     11.53.30 JOB09860  $HASP1301 TEST6 in job group TESTE1 queued for execu
SC74     11.53.30 JOB09857  $HASP373 TEST3    STARTED - INIT 2    - CLASS A
SC74     11.53.30 JOB09861  $HASP1300 TEST7 registered to job group TESTE1
SC74     11.53.30 JOB09857  $HASP395 TEST3    ENDED - RC=0000
SC74     11.58.02 JOB09860  $HASP373 TEST6    STARTED - INIT 1    - CLASS A
SC74     11.58.02 JOB09860  $HASP395 TEST6    ENDED - RC=0000
SC74     11.58.02 JOB09861  $HASP1301 TEST7 in job group TESTE1 queued for execu
SC74     11.58.02 JOB09856  $HASP1301 TEST2 in job group TESTE1 queued for execu
SC74     11.58.02 JOB09861  $HASP373 TEST7    STARTED - INIT 1    - CLASS A
SC74     11.58.02 JOB09856  $HASP373 TEST2    STARTED - INIT 2    - CLASS A
SC74     11.58.02 JOB09861  $HASP395 TEST7    ENDED - RC=0000
SC74     11.58.02 JOB09856  $HASP395 TEST2    ENDED - RC=0000
SC74     11.58.02 JOB09859  $HASP1301 TEST5 in job group TESTE1 queued for execu
         11.58.02 JOB09858  $HASP1301 TEST4 in job group TESTE1 queued for execu
         11.58.02 JOB09858  $HASP373 TEST4    STARTED - INIT 2    - CLASS A
         11.58.02 JOB09859  $HASP373 TEST5    STARTED - INIT 1    - CLASS A
         11.58.02 JOB09858  $HASP395 TEST4    ENDED - RC=0000
         11.58.02 JOB09859  $HASP395 TEST5    ENDED - RC=0000
         11.58.02 G0009854  $HASP1304 job group TESTE1 is complete
       JES2 JOB STATISTICS ------
           11 CARDS READ
           25 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
            2 SYSOUT SPOOL KBYTES
         0.00 MINUTES EXECUTION TIME
        1 //TESTE1   JOBGROUP
        2 //TEST1    GJOB
        3 //         BEFORE  NAME=(TEST2,TEST3)
        4 //TEST2    GJOB
        5 //         BEFORE  NAME=(TEST4,TEST5)
        6 //TEST3    GJOB
        7 //TEST4    GJOB
        8 //TEST5    GJOB
        9 //TEST6    GJOB
       10 //         BEFORE  NAME=(TEST7,TEST2)
       11 //TEST7    GJOB
       12 //TESTE1   ENDGROUP
       IO. MESSAGE
       12 HASP1111 JOBGROUP is valid
```

Figure D-6 JOBGROUP Messages

Comparing both job stream executions (as shown in Figure D-2 on page 228 and Figure D-6 on page 232), you can see that the jobstreams executed the same way in both cases. The JOBGROUP that was created by JES2 when processing the //* NET JES3 LECL statements controlled the execution of the jobs in the job stream, such as the JOBGROUP we created with the corresponding SCHEDULE JCL statements.

E

SPOOL partitioning exits sample code

This appendix contains sample code for exits 11 and 12 that can be used to control the spool partitioning features on JES2. These features can be useful for users who are looking for a solution to JES2 spool shortage condition or for JES3 users that use the JES3 spool partitioning function during migration from JES3 to JES2.

The spool partitioning allows you to isolate different types of spool data. Isolating spool data in separate partitions can help you improve spool performance, spool recovery procedures, and spool space management.

This appendix includes the following topics:

E.1 Sample exits overview

The spool partitioning that is controlled by these samples are based on RACF FACILITY class profiles. These profiles are used to control the users and jobs that can use spool partitioning and the volumes for spooling the sysouts that are produced by these jobs and, if authorized, use more JES2 spool volumes to overflow the sysouts.

Before implementing this exit, you must determine if your installation uses spool partitioning. Your installation uses spool partitioning if FENCE=ACTIVE=YES is specified on the SPOOLDEF initialization statement.

These exits are used as listed in Table E-1.

Table E-1 Comparison of exits 11 and 12

	Exit 11	Exit 12
Spool partitioning mask	► Initializes and resets bits in the mask. ► Can be used to define spool partitioning for the job.	Can only reset bits in the mask to allow spool space to be allocated from more spool volumes.
Started to	Allocate spool space for the first time for the job.	Allocate more spool space when JES2 determines that the spools that use the allowed mask of the job must be expanded.

When exit 11 is called for the first time, situations exist that it are called again if the following conditions are met:

- The job was not assigned the maximum number of spool volumes (SPOOLDEF FENCE=VOLUMES=nnnn), regardless of whether space is available on the spool volumes from which the job is permitted to allocate space.
- The job assigned the maximum number of volumes and no space is available for allocation (that is, the volumes are full, the volumes are not available for allocation, or the volumes do not have affinity for the system).

Exit 12 is taken when JES2 determines that the spools that are using the allowed mask for the job that was set by exit 11 must be updated. The spools that are using the allowed mask are updated in the following situations:

- The job is not yet using the maximum number of spool volumes (SPOOLDEF FENCE=VOLUMES=nnnn), regardless of whether space is available on the spool volumes from which the job is permitted to allocate space.
- The job is using the maximum number of volumes (CCTFNCNT in HCCT) and no space is available for allocation (that is, the volume is full, the volume is not available for allocation, or the volume does not have affinity for the system) on the spool volumes from which the job is permitted to allocate space.

Defining spool partitions

To define spool partitions that use the sample exits, define the RACF $JES2.SPART.VOL.sysid.volid profiles on class FACILITY and grant READ access to users that you want to use the partition. The partition can be defined by using a partial spool volume name on RACF profile.

Defining spool partition overflow

To provide for instances when a requested spool partition is full, you can specify where each spool partition's overflow data is sent. To make this specification, use the RACF $JES2.SPART.OVRFL.sysid.volid profiles on FACILITY class and grant READ access to users that you want use these volumes to overflow spool data.

Also, to allow a job to overflow spool data from a partition, you must define a RACF $JES2.SPART.EXT.sysid.jobname profile on class FACILITY and grant READ access to users extend the current partition to volumes on overflow partition.

Defining default partition

If you want to prevent the JES2 from using all of the available spool data, you must define volumes to a default partition by defining a RACF $JES2.SPART.DFLT.sysid.volid profile on class FACILITY. You also must grant READ access to all users that are allowed to use the spool partitioning process that is provided by exits in this sample.

The volumes that are assigned to a default partitions are used only if all primary and overflow volumes become full and the job requires more spool space to processing.

If the volumes on default partition also becomes full, the exit 12 sends a message to the operator requesting to retry the process or cancel the processing of spool partitioning and allocate spool data into unassigned volumes with space available.

Define at least one volume for the default partition to receive data from overflow volumes without affecting the reserved spool space. If you do not define a default partition for the exits, JES2 uses the reserved spool space that is not assigned to the exits as normal processing if all volumes are full.

E.1.1 RACF profiles used by exits

To implement and control spool partitioning functionality, the exits use RACF profiles in the FACILITY class in accordance with the authorization and qualification requirements of the required spooling resources.

The partitioning control that is offered by these exits is based on the job name and type of address space (which differs from the partitioning control that is provided by JES3 based on the job execution class). Therefore, to permit the jobs to use the spool partitioning functionality that is provided by exits, the following profiles must be defined to RACF:

- `$JES2.SPART.jobtype.sysid.jobname`

  This profile is used by exit 11 to control by jobnames the users that can use the provided spool partitioning functionality. The following variables are used on this profiles:

  - `jobtype`: The type of job that is authorized to use the spool partitioning functionality (JOB, TSU, or STC).
  - `sysid`: The system ID of the MAS member where the spool partitioning functionality is active to the specific job.
  - `jobname`: The name partial or fully qualified of job authorized to use the spool partitioning functionality that is provided by exits.

- `$JES2.SPART.VOL.sysid.volid`

  This profile is used by exit 11 or exit 12 to identify the spool volumes that can be used as partition space to hold the job data that is based on partial or fully qualified volume name. The following variables are used on this profiles:

  – `sysid`: The system ID of the MAS member where the spool volume is used as available to spool partitioning.

  – `volid`: The name of the spool volume that can be used as spool space for partitioning the spool data that is requested by a job.

- `$JES2.SPART.EXT.sysid.jobname`

  This profile is used by exit 12 to identify when no other space is available on volumes that are used to hold the original spool data to a job if the spool data can be overflowed to other volumes. The following variables are used on this profile:

  – `sysid`: The system ID of the MAS member where the extra space to the job is provided, if available.

  – `jobname`: The name of job that can use the spool partitioning overflow process that is provided by the exits when all volumes on the primary spool partition are full.

- `$JES2.SPART.OVRFL.sysid.volid`

  This profile is used by exit 11 and 12 to identify the volumes that are available to receive overflow data from jobs that started to write data in a different volume that is full. The volumes that are identified on this profile can be also identified on the original spool partitioning profile. The following variables are used on this profile:

  – `sysid`: The system ID of the MAS member where the spool volume is used as available to receive spool overflow data.

  – `volid`: The name of spool volume that can be used as spool space for overflow of original spool data.

- `$JES2.SPART.DFLT.sysid.volid`

  This profile is used by exit 11 and 12 to identify the volumes that are available to receive overflowed data from jobs that started to write data in a different volume that are full. The volumes that are identified on this profile can be also identified on the original spool partitioning profile. The following variables are used on this profile:

  – `sysid`: The system ID of the MAS member where the spool volume is used as available to receive spool overflow data.

  – `volid`: The name of the spool volume that can be used as spool space for the overflow of original spool data.

- `$JES2.SPART.CLASS.sysid.jobclass`

  This profile is used by exit 11 and 12 to control by jobclass definition the users that can use spool partitioning. The following variables are used on this profile:

  – `sysid`: The system ID of the MAS member where the spool partitioning functionality is active to the specific job.

  – `jobclass`: The 1 - 8 characters class definition that is authorized to use the spool partitioning functionality that is provided by exits.

Sample profile definitions

How you can implement JES3 spool partitioning and migrate it to JES2 with the sample exits 11 and 12 is shown in Example E-1.

Example E-1 Sample JES3 spool partitioning definitions

```
SPART,NAME=NORMAL,DEF=YES
SPART,NAME=SPECIAL,OVFL=SPARE
SPART,NAME=SPARE

TRACK,DDNAME=SPOOL1,SPART=NORMAL
TRACK,DDNAME=SPOOL2,SPART=NORMAL
TRACK,DDNAME=SPOOL3,SPART=NORMAL
TRACK,DDNAME=SPOOL4,SPART=SPECIAL
TRACK,DDNAME=SPPOL5,SPART=SPARE

CLASS,NAME=A,SPART=NORMAL
CLASS,NAME=B,SPART=SPECIAL
SYSOUT,CLASS=X,SPART=NORMAL
SYSOUT,CLASS=Y,SPART=SPECIAL
```

With the new APAR OA55792 applied, the JES2 supports the use of job class and message class to select the jobs that can use spool partitioning. Because the APAR was not available during the tests, the version that is presented uses the JOBNAME to select the candidate jobs. Also, you must permit special users to use the spool partitioning function, as shown in Example E-2.

Example E-2 Sample RACF profile definitions

```
$JES2.SPART.jobtype.sysid.jobname

   $JES2.SPART.JOB.*.EMG*: Permit special users for processing batch jobs by using
   spool partitioning.

$JES2.SPART.VOL.sysid.volid

   $JES2.SPART.VOL.*.SPOOL1: Permit normal users

   $JES2.SPART.VOL.*.SPOOL2: Permit normal users

   $JES2.SPART.VOL.*.SPOOL3: Permit normal users

   $JES2.SPART.VOL.*.SPOOL4: Permit special users to allocate spool space for batch
   job processing.

$JES2.SPART.EXT.sysid.jobname

   $JES2.SPART.EXT.*.EMG*: Permit special users to acquire more spool space if the
   primary partition became full

$JES2.SPART.CLASS.sysid.class

   $JES2.SPART.*.A*: Permit that jobs running with jobclasses started with A are
   able to use spool partitioning.

$JES2.SPART.OVRFL.sysid.volid

   $JES2.SPART.OVLF.*.SPOOL5: Permit special users to use the spool volume SPOOL5
   as overflow space from primary partition.

$JES2.SPART.DFLT.sysid.volid
```

The following profile definitions permit all users with access to use spool partitioning to allocate space on the spool volume:

```
$JES2.SPART.DFLT.*.SPOOL1 UACC(READ)
```

```
$JES2.SPART.DFLT.*.SPOOL2 UACC(READ)
```

```
$JES2.SPART.DFLT.*.SPOOL3 UACC(READ)
```

E.2 Exit 11 program source code

The sample code for JES2X011 exit that is used to provide the spool partitioning process to JES2 is shown in Example E-3.

Example E-3 JES2 exit 11 code sample

```
TITLE 'JES2 EXIT011 - SPOOL PARTITIONING'
*/*******************************************************************/*
*/* PROGRAM  - JES2X011                                             */*
*/*                                                                 */*
*/* FUNCTION - THIS EXIT IS DESIGNED TO PROVIDE SPOOL PARTITIONING  */*
*/*           FUNCTIONS TO JES2 SIMILAR TO JES3                     */*
*/*                                                                 */*
*/*******************************************************************/*
         EJECT
         PRINT GEN
*/*******************************************************************/*
*/* COPY OF JES2 $HASPGBL MAPPING                                   */*
*/*******************************************************************/*
         COPY  $HASPGBL
         EJECT
*/*******************************************************************/*
*/* JES2 MACRO  $MODULE EXPANSION                                   */*
*/*******************************************************************/*
JES2X011 $MODULE ENVIRON=JES2,                                         X
               RMODE=ANY,                                              X
               IBMJES2=SAMPLE,                                         X
               $BUFFER,              REQ BY $REQBUF, $FREEBUF          X
               $CAT,                 REQ BY HCT                        X
               $CNVWORK,             CONV. PROCESSOR PCE WORK1 AREA    X
               $DAS,                 IOT MAPPINT MACRO                 X
               $DTE,                 REQ BY PCE                        X
               $DTECNV,              REQ BY DTE                        X
               $ERA,                 REQ BY DTE                        X
               $HASPEQU,             HASP EQUATES                      X
               $HCCT,                REQ BY $SAVE, $RETURN, ETC        X
               $HCT,                 REQ BY $SAVE, $RETURN, ETC        X
               $IOT,                 IOT MAPPINT MACRO                 X
               $JCT,                 REQ BY CAT                        X
               $JCTX,                REQ BY CAT                        X
               $JQE,                 REQ BY HCT                        X
               $MIT,                 REQ BY MODEND                     X
               $PADDR,               REQ BY HCT                        X
               $PCE,                 REQ BY HCT                        X
               $SCAT,                REQ BY HCT                        X
               $TAB,                 REQ BY $CNVWORK                   X
               $TRE,                 REQ BY $JCTXGET                   X
               $TQE,                 REQ BY $CNVWORK                   X
               $XECB,                REQ BY DTE                        X
               $XIT,                                                   X
               CVT,                  COMMUNICATION VECTOR TABLE        X
               DEB,                  DATA EXTEND BLOCK                 X
               CNMB,                 CONVERTER MESSAGE BUFFER          X
               RPL,                  ACB REQUEST PARAMETER LIST        X
               SDWA,                 SYSTEM DIAGNOSIS WORK1ING AREA    X
               WPL                   REQ BY $$WTO
*/*******************************************************************/*
*/* JES2 EXIT 11 ENTRY POINT                                        */*
*/*******************************************************************/*
EXIT011  $ENTRY BASE=R12,CSECT=YES
         SAVE  (14,12)
         LR    R12,R15
         LM    R7,R9,0(R1)
         USING IOT,R7
         USING JCT,R8
         USING HCT,R11
         L     R2,$HCCT
         USING HCCT,R2
*/*******************************************************************/*
*/* OBTAIN STORAGE TO VARIABLES USED BY EXIT                        */*
*/*******************************************************************/*
STORAGE_OBTAIN EQU *
         STORAGE OBTAIN,LENGTH=WORKLEN,LOC=31
         ST    R13,4(R1)
         ST    R1,8(R13)
         LR    R13,R1
         USING WORKAREA,R13
*/*******************************************************************/*
*/* TEST IF JCT IS PRESENT AND IF THE MAIN CONDITIONS TO PROCESS    */*
*/*******************************************************************/*
TEST_EXIT_CALL EQU *
         XC    RETURN_CODE,RETURN_CODE
         LTR   R8,R8
         BZ    END_OF_EXIT
         TM    JCTUSERB,X'B0'
         BO    END_EXIT_08
*/*******************************************************************/*
*/* IDENTIFY THE TYPE OF JCT (TSO, JOB OR STC)                      */*
*/*******************************************************************/*
         MVC   RACF_JOB_TYPE,RACF_PROF_JOB
         TM    JCTJOBFL,JCTBATCH
         BO    VALID_JOB_TYPE
         MVC   RACF_JOB_TYPE,RACF_PROF_TSU
         TM    JCTJOBFL,JCTTSUJB
         BO    VALID_JOB_TYPE
         MVC   RACF_JOB_TYPE,RACF_PROF_STC
         TM    JCTJOBFL,JCTSTCJB
         BNO   END_OF_EXIT
```

```
*/*******************************************************************/*
*/* GET USERID AND SYSID TO IDENTIFY THE JOB AND PROCESSING MEMBER  */*
*/*******************************************************************/*
VALID_JOB_TYPE EQU *
         PERFORM GET_SYSID,R
         PERFORM GET_USERID,R
*/*******************************************************************/*
*/* TEST IF FENCE IS ACTIVE TO THIS JES2 MEMBER                     */*
*/*******************************************************************/*
         TM    $FLAG1,$MVFENCE
         BO    GET_REQUESTED_INFO
         GETMSG 6,WTO_MESSAGE,MESSAGES
         MVC   WTO_MESSAGE+MSG6_SYSID-MSG6(L'SYSID),SYSID
         PERFORM SEND_WTO,R
         B     END_OF_EXIT
*/*******************************************************************/*
*/* GET RQQUIRED INFORMATION FROM JCT, IOT AND HCT                  */*
*/*******************************************************************/*
GET_REQUESTED_INFO EQU *
         XC    FLAG,FLAG
         MVC   #SPOOL_VOLUMES,$SPOLNUM
         MVC   #FENCE_VOLUMES,$FNCCNT
         MVC   DAS_ADDRESS,$DASAREA
         MVC   DAS_FIRST,$DASFRST
         MVC   JOB_NAME,JCTJNAME
         MVC   JOB_NUMBER,JCTJBNUM
         MVC   JOB_JOBID,JCTJOBID
         MVC   JOB_CLASS,JCTJCLAS
         MVC   SPOOL_ALLOCATED_MASK,IOTSPMSK
         MVC   SPOOL_AVAILABLE_MASK,CCTVBLOB
         XC    SPOOL_ALLOWED_MASK,SPOOL_ALLOWED_MASK
         XC    SPOOL_VOL_SET,SPOOL_VOL_SET
*/*******************************************************************/*
*/* VERIFY IF THE JOBNAME IS A CANDIDATE TO USE SPOOL PARTITIONING  */*
*/*******************************************************************/*
VALIDATE_JOBNAME EQU *
         CLEAR RACF_PROFILE
         MVC   RACF_TYPE(L'RACF_PROF_TYPE),RACF_PROF_TYPE
         MVC   RACF_SPOOLJ(L'RACF_JOB_TYPE),RACF_JOB_TYPE
         MVI   RACF_SPOOLJ_DOT1,C'.'
         MVC   RACF_SPOOLJ_SYSID(L'SYSID),SYSID
         MVI   RACF_SPOOLJ_DOT2,C'.'
         MVC   RACF_SPOOLJ_JOB(L'JOB_NAME),JOB_NAME
         PERFORM RACF_CHECK_AUTH,R
         LTR   R15,R15
         BNZ   END_OF_EXIT
         OI    FLAG,FLAG_JOBNAME
*/*******************************************************************/*
*/* SEND A MESSAGE INDICATING THAT JOB IS VALID CANDIDATE           */*
*/*******************************************************************/*
         GETMSG 1,WTO_MESSAGE,MESSAGES
         MVC   WTO_MESSAGE+MSG1_SYSID-MSG1(L'SYSID),SYSID
         MVC   WTO_MESSAGE+MSG1_JOBNAME-MSG1(L'JOB_NAME),JOB_NAME
       MVC WTO_MESSAGE+MSG1_JOBTYPE-MSG1(L'RACF_JOB_TYPE),RACF_JOB_TYPE
         PERFORM SEND_WTO,R
         MVC   SPOOL_VOL_TYPE,VOL_ALLOWED
*/********************************************************************/*
*/* VALIDATE THE $DAS HEADER ON $IOT DATA AREA                       */*
*/********************************************************************/*
START_DAS_VALIDATION EQU *
         L     R1,$DASAREA
         CLC   DAS_POOL_ID,0(R1)
         BE    START_DAS_SEARCH
         GETMSG 2,WTO_MESSAGE,MESSAGES
         PERFORM SEND_WTO,R
         B     END_OF_EXIT
*/********************************************************************/*
*/* START THE $DAS SEARCHING PTOCESS TO FIND SPOOL VOLUMES           */*
*/********************************************************************/*
START_DAS_SEARCH EQU *
         L     R3,$DASFRST
         USING DAS,R3
*/********************************************************************/*
*/* INITIALIZE VARIABLES AND COUNTERS                                */*
*/********************************************************************/*
         LA    R1,SPOOL_ALLOWED_MASK
         ST    R1,SPOOL_MASK_ADDRESS
         LA    R1,SPOOL_AVAILABLE_MASK
         ST    R1,SPOOL_AVAIL_ADDRESS
         XC    BITMASK,BITMASK
         OI    BITMASK,X'80'
         LH    R4,#SPOOL_VOLUMES
*/********************************************************************/*
*/* COMPARE THE TOTAL VOLUME ADDED WITH FENCE VALUE                  */*
*/********************************************************************/*
NEW_DAS_ENTRY EQU *
         CLC   SPOOL_VOL_SET,#FENCE_VOLUMES
         BNL   END_DAS_CHAIN
*/********************************************************************/*
*/* VALIDATE IF THE SPOOL VOLUME IS AVAILABLE FOR ALLOCATION         */*
*/********************************************************************/*
         TM    DASFLAG,DASACTIV
         BNO   NEXT_DAS_ENTRY
*/********************************************************************/*
*/* TESTS IF THE VOLUME HAVE SPACE AVAILABLE TO BE ALLOCATED         */*
*/********************************************************************/*
         L     R1,SPOOL_AVAIL_ADDRESS
         XR    R15,R15
         IC    R15,BITMASK
         EX    R15,TEST_BITMASK
         BNO   NEXT_DAS_ENTRY
*/********************************************************************/*
*/* GET VOLUME ID FROM $DAS AND CHECK RACF PROFILE ACCESS            */*
*/********************************************************************/*
         MVC   SPOOL_VOLUME,DASVOLID
         PERFORM CHECK_VOLUME_ACCESS,R
         PERFORM RACF_CHECK_AUTH,R
         LTR   R15,R15
         BNZ   NEXT_DAS_ENTRY
*/********************************************************************/*
```

```
*/* ADD THE SPOOL VOLUME AS A VOLUME ALLOWED TO BE ALLOCATED        */*
*/*******************************************************************/*
ADD_SPOOL_VOLUME EQU *
         XR    R1,R1
         IC    R1,SPOOL_VOL_SET
         LA    R1,1(R1)
         STC   R1,SPOOL_VOL_SET
         OI    FLAG,FLAG_SPOOL
*/*******************************************************************/*
*/* SET A BIT ON BITMASK TO PUT A VOLUME AS ALLOWED                 */*
*/*******************************************************************/*
         L     R1,SPOOL_MASK_ADDRESS
         OC    0(L'BITMASK,R1),BITMASK
*/*******************************************************************/*
*/* SEND MESSAGE TO CONSOLE WITH SPOOL VOLUME ADDED TO JOB          */*
*/*******************************************************************/*
         GETMSG 3,WTO_MESSAGE,MESSAGES
         MVC  WTO_MESSAGE+MSG3_VOLUME-MSG3(L'SPOOL_VOLUME),SPOOL_VOLUME
         MVC   WTO_MESSAGE+MSG3_JOBNAME-MSG3(L'JOB_NAME),JOB_NAME
       MVC WTO_MESSAGE+MSG3_JOBTYPE-MSG3(L'RACF_JOB_TYPE),RACF_JOB_TYPE
     MVC WTO_MESSAGE+MSG3_VOLTYPE-MSG3(L'SPOOL_VOL_TYPE),SPOOL_VOL_TYPE
         PERFORM SEND_WTO,R
*/*******************************************************************/*
*/* JUMP TO NEXT AVAILABLE DAS ENTRY ON DAS CHAIN                   */*
*/*******************************************************************/*
NEXT_DAS_ENTRY EQU *
         XR    R1,R1
         ICM   R1,15,DASTRAKQ
         LTR   R1,R1
         BZ    END_DAS_CHAIN
         LR    R3,R1
         A     R3,$DASAREA
         PERFORM NEXT_BITMASK,R
         BCT   R4,NEW_DAS_ENTRY
*/*******************************************************************/*
*/* CHECK IF WAS SET ANY VOLUME TO BE USED BY JOB AND SET JCTUSER  */*
*/*******************************************************************/*
END_DAS_CHAIN EQU *
         TM    FLAG,FLAG_SPOOL
         BNO   SEARCH_OVERFLOW_VOLUMES
         MVC   0(L'SPOOL_ALLOWED_MASK,R9),SPOOL_ALLOWED_MASK
         OI    JCTUSERB,X'B0'
         B     END_EXIT_08
*/*******************************************************************/*
*/* SET TO SEARCH FOR DEFAULT VOLUMES TO BE USED BY JOB             */*
*/*******************************************************************/*
SEARCH_OVERFLOW_VOLUMES EQU *
         TM    FLAG,FLAG_OVERFLOW
         BO    SEARCH_DEFAULT_VOLUMES
         OI    FLAG,FLAG_OVERFLOW
         MVC   SPOOL_VOL_TYPE,VOL_OVERFLOW
         B     START_DAS_VALIDATION
*/*******************************************************************/*
*/* SET TO SEARCH FOR DEFAULT VOLUMES TO BE USED BY JOB             */*
*/*******************************************************************/*
```

```
SEARCH_DEFAULT_VOLUMES EQU *
         TM    FLAG,FLAG_DEFAULT
         BO    NO_VOLUMES_FOUND
         OI    FLAG,FLAG_DEFAULT
         MVC   SPOOL_VOL_TYPE,VOL_DEFAULT
         B     START_DAS_VALIDATION
*/****************************************************************/*
*/* SEND MESSAGE WITH NO VOLUMES FOUND CONDITION AND RETURN TO JES  */*
*/****************************************************************/*
NO_VOLUMES_FOUND EQU *
         GETMSG 4,WTO_MESSAGE,MESSAGES
         MVC   WTO_MESSAGE+MSG4_JOBNAME-MSG4(L'JOB_NAME),JOB_NAME
         PERFORM SEND_WTO,R
         B     END_OF_EXIT
*/****************************************************************/*
*/* POINT TO NEXT BITMAKS TO BE USED FOR SPOOL VOLUME            */*
*/****************************************************************/*
NEXT_BITMASK EQU *
         TM    BITMASK,X'01'
         BO    SHIFT_SPOOL_MASK
         XR    R1,R1
         IC    R1,BITMASK
         SRL   R1,1
         STC   R1,BITMASK
         BR    R10
*/****************************************************************/*
*/* WALK THRU SPOOL_ALLOWED_MASK AND SPOOL_AVAILABLE_MASK         */*
*/****************************************************************/*
SHIFT_SPOOL_MASK EQU *
         L     R1,SPOOL_MASK_ADDRESS
         LA    R1,L'BITMASK(R1)
         ST    R1,SPOOL_MASK_ADDRESS
         L     R1,SPOOL_AVAIL_ADDRESS
         LA    R1,L'BITMASK(R1)
         ST    R1,SPOOL_AVAIL_ADDRESS
         XC    BITMASK,BITMASK
         OI    BITMASK,X'80'
         BR    R10
*/****************************************************************/*
*/* VALIDATE THE SPOOL VOLUME AGAINST RACF PROFILE ACCESS         */*
*/****************************************************************/*
CHECK_VOLUME_ACCESS EQU *
         CLEAR RACF_PROFILE
         MVC   RACF_TYPE(L'RACF_PROF_TYPE),RACF_PROF_TYPE
         TM    FLAG,FLAG_OVERFLOW
         BO    CHECK_OVERFLOW_VOLUME
         TM    FLAG,FLAG_DEFAULT
         BO    CHECK_DEFAULT_VOLUME
*/****************************************************************/*
*/* SEARCH RACF PROFILE TO VALIDATE THE ACCESS TO SPOOL VOLUME     */*
*/****************************************************************/*
         MVC   RACF_SPOOLV(L'RACF_PROF_SPOOLV),RACF_PROF_SPOOLV
         MVC   RACF_SPOOLV_SYSID(L'SYSID),SYSID
         MVI   RACF_SPOOLV_DOT,C'.'
         MVC   RACF_SPOOLV_VOLUME(L'SPOOL_VOLUME),SPOOL_VOLUME
         BR    R10
*/*******************************************************************/*
*/* SEARCH RACF PROFILE TO VALIDATE THE ACCESS TO DEFAULT VOLUME    */*
*/*******************************************************************/*
CHECK_OVERFLOW_VOLUME EQU *
         MVC   RACF_SPOOLO(L'RACF_PROF_SPOOLO),RACF_PROF_SPOOLO
         MVC   RACF_SPOOLO_SYSID(L'SYSID),SYSID
         MVI   RACF_SPOOLO_DOT,C'.'
         MVC   RACF_SPOOLO_VOLUME(L'SPOOL_VOLUME),SPOOL_VOLUME
         BR    R10
*/*******************************************************************/*
*/* SEARCH RACF PROFILE TO VALIDATE THE ACCESS TO DEFAULT VOLUME    */*
*/*******************************************************************/*
CHECK_DEFAULT_VOLUME EQU *
         MVC   RACF_SPOOLD(L'RACF_PROF_SPOOLD),RACF_PROF_SPOOLD
         MVC   RACF_SPOOLD_SYSID(L'SYSID),SYSID
         MVI   RACF_SPOOLD_DOT,C'.'
         MVC   RACF_SPOOLD_VOLUME(L'SPOOL_VOLUME),SPOOL_VOLUME
         BR    R10
*/*******************************************************************/*
*/* ROUTINE TO GET SYSID FROM SYSTEM                                */*
*/*******************************************************************/*
GET_SYSID EQU *
         L     R1,CVTPTR
         L     R1,CVTSMCA-CVTMAP(R1)
         USING SMCABASE,R1
         MVC   SYSID,SMCASID
         BR    10
*/*******************************************************************/*
*/* VERIFY IF EXISTS USER PARAMETER ON JOB CARD OR GET A USER       */*
*/*******************************************************************/*
GET_USERID EQU *
         CLEAR JOB_USERID
         CLI   JCTNOUSR,X'00'
         BNE   USERID_FOUND
         GETMSG 5,WTO_MESSAGE,MESSAGES
         MVC   WTO_MESSAGE+MSG5_JOBNAME-MSG5(L'JOB_NAME),JOB_NAME
         PERFORM SEND_WTO,R
         B     END_OF_EXIT
*/*******************************************************************/*
*/* GET THE USERID FROM JCT SUBMITTING USER                         */*
*/*******************************************************************/*
USERID_FOUND EQU *
         OI    FLAG,FLAG_USERID
         MVC   JOB_USERID(8),JCTNOUSR
         TM    JCTFLAG1,JCT1UNDF
         BOR   R10
         CLC   JCTJUSID,=8X'00'
         BER   R10
         CLC   JCTJUSID,JCTNOUSR
         BER   R10
         MVC   JOB_USERID(8),JCTJUSID
         BR    R10
*/*******************************************************************/*
*/* SUBROUTINE TO REQUEST RACF ACCESS VALIDATION                    */*
```

```
*/********************************************************************/*
RACF_CHECK_AUTH EQU *
         STM   R3,R4,SAVE34
         MVC   RACFT(RACLEN),RAC_LIST
         LA    R3,RACF_PROFILE
         LA    R4,RACF_CLASS_FACILITY
         RACROUTE REQUEST=AUTH,                                        X
               WORKA=RACWORK,                                          X
               ENTITY=((3)),                                           X
               CLASS=((4)),                                            X
               ATTR=READ,                                              X
               GENERIC=ASIS,                                           X
               USERID=JOB_USERID,                                      X
               RELEASE=7790,                                           X
               LOG=NONE,                                               X
               MF=(E,RACFT)
         LM    R3,R4,SAVE34
         BR    10
*/********************************************************************/*
*/* SUBROUTINE TO SEND A MESSAGE TO CONSOLE                          */*
*/********************************************************************/*
SEND_WTO EQU  *
         MVC   WTO_MSGL,=AL2(L'WTO_MESSAGE)
         MVC   WTO_EXEC,WTO_LIST
         $$WTO WTO_EXEC,TEXT=WTO_MSG
         BR    R10
*/********************************************************************/*
*/* END OF EXIT - RELEASE ACQUIRED STORAGE AND RETURN TO CALLER      */*
*/********************************************************************/*
END_EXIT_08 EQU *
         MVC   RETURN_CODE,FULL_8
END_OF_EXIT EQU *
         L     R15,RETURN_CODE
         LR    R1,R13
         L     R13,4(R13)
         ST    R15,16(R13)
         STORAGE RELEASE,LENGTH=WORKLEN,ADDR=(1)
         LM    R14,R12,12(R13)
         BR    R14
*/********************************************************************/*
*/* INSTRUCTION AREA USED TO EXECUTE                                 */*
*/********************************************************************/*
TEST_BITMASK TM 0(R1),X'00'
*/********************************************************************/*
*/* WORKAREA OBTAINED BY EXIT                                        */*
*/********************************************************************/*
WORKAREA       DSECT
SAVEAREA       DS 18F
SAVE34         DS 2F
DOUBLE         DS D
RETURN_CODE    DS F
BITMASK        DS X
#SPOOL_VOLUMES DS H
#FENCE_VOLUMES DS X
JOB_NUMBER     DS F
```

```
JOB_NAME       DS CL8
JOB_USERID     DS CL8
JOB_CLASS      DS CL1
JOB_JOBID      DS CL8
SYSID          DS CL4
DAS_ADDRESS    DS F
DAS_FIRST      DS F
SPOOL_MASK_ADDRESS   DS F
SPOOL_AVAIL_ADDRESS  DS F
SPOOL_ALLOCATED_MASK DS 8F
SPOOL_AVAILABLE_MASK DS 8F
SPOOL_ALLOWED_MASK   DS 8F
SPOOL_VOLUME         DS CL6
SPOOL_VOL_SET        DS X
SPOOL_VOL_TYPE       DS CL7
FLAG           DS X
FLAG_CLASS    EQU X'01'
FLAG_SPOOL    EQU X'02'
FLAG_USERID   EQU X'04'
FLAG_JOBNAME  EQU X'08'
FLAG_CANCEL   EQU X'10'
FLAG_OVERFLOW EQU X'20'
FLAG_DEFAULT  EQU X'40'
*/*******************************************************************/*
*/* DEFINE AREA TO MAP RACF PROFILES TO BE USED                     */*
*/*******************************************************************/*
RACF_JOB_TYPE     DS CL3
RACF_PROFILE      DS CL44
         ORG RACF_PROFILE
RACF_TYPE         DS CL(L'RACF_PROF_TYPE)
RACF_TYPE_VAR     DS 0C
         ORG RACF_TYPE_VAR
RACF_SPOOLJ       DS CL(L'RACF_JOB_TYPE)
RACF_SPOOLJ_DOT1  DS CL1
RACF_SPOOLJ_SYSID DS CL4
RACF_SPOOLJ_DOT2  DS CL1
RACF_SPOOLJ_JOB   DS CL8
         ORG RACF_TYPE_VAR
RACF_SPOOLV       DS CL(L'RACF_PROF_SPOOLV)
RACF_SPOOLV_SYSID DS CL4
RACF_SPOOLV_DOT   DS CL1
RACF_SPOOLV_VOLUME DS CL6
         ORG
         ORG RACF_TYPE_VAR
RACF_SPOOLO       DS CL(L'RACF_PROF_SPOOLO)
RACF_SPOOLO_SYSID DS CL4
RACF_SPOOLO_DOT   DS CL1
RACF_SPOOLO_VOLUME DS CL6
         ORG
         ORG RACF_TYPE_VAR
RACF_SPOOLD       DS CL(L'RACF_PROF_SPOOLD)
RACF_SPOOLD_SYSID DS CL4
RACF_SPOOLD_DOT   DS CL1
RACF_SPOOLD_VOLUME DS CL6
         ORG
```

```
*/*******************************************************************/*
*/* DEFINE CLASS TO USE WITH RACF AND LIST FORM OF MACRO RACROUTE   */*
*/*******************************************************************/*
         DS    0F
RACWORK  DS    CL512                   WORK AREA DO RACF
         DS    0F
RACFT    RACROUTE REQUEST=AUTH,                                        X
               WORKA=*-*,                                              X
               CLASS='FACILITY',                                       X
               ATTR=READ,                                              X
               RELEASE=7790,                                           X
               MF=L
*/*******************************************************************/*
*/* DEFINE WORK AREA TO BE USED BY WTO MACRO                        */*
*/*******************************************************************/*
WTO_EXEC DS    CL(WTO_LEN)
WTO_MSG  DS    0H
WTO_MSGL DS    AL2
WTO_MESSAGE DS CL100
*/*******************************************************************/*
*/* DEFINE AREA USED BY PERFORM PROCESS                             */*
*/*******************************************************************/*
         PERFORM GENERATE
WORKLEN  EQU   *-WORKAREA
         EJECT
*/*******************************************************************/*
*/* USED CONSTANTS BY EXIT                                          */*
*/*******************************************************************/*
EXIT011 CSECT
EXIT011 AMODE 31
EXIT011 RMODE ANY
BINZEROS       DC 2F'0'
FULL_0         DC F'0'
FULL_4         DC F'4'
FULL_8         DC F'8'
FULL_MASK      DC 32X'FF'
DAS_POOL_ID    DC CL13'**DAS  POOL**'
VOL_DEFAULT    DC C'DEFAULT'
VOL_ALLOWED    DC C'ALLOWED'
VOL_OVERFLOW   DC C'OVRFLOW'
*/*******************************************************************/*
*/* DEFINE CONSTANTS TO BE USED FOR RACF PROCESSING                 */*
*/*******************************************************************/*
RACF_CLASS_FACILITY DC AL1(L'FACILITY)
FACILITY            DC C'FACILITY'
RACF_PROF_TYPE    DC  C'$JES2.SPART.'
RACF_PROF_JOB     DC  C'JOB'
RACF_PROF_TSU     DC  C'TSU'
RACF_PROF_STC     DC  C'STC'
RACF_PROF_SPOOLV  DC  C'VOL.'
RACF_PROF_SPOOLO  DC  C'OVRFL.'
RACF_PROF_SPOOLD  DC  C'DFLT.'
         DS    0F
RAC_LIST RACROUTE REQUEST=AUTH,                                        X
               WORKA=*-*,                                              X
               CLASS='FACILITY',                                        X
               ATTR=READ,                                               X
               RELEASE=7790,                                            X
               MF=L
RACLEN   EQU   (*-RAC_LIST)
*/********************************************************************/*
*/* DEFINE LIST FORM TO $$WTO MACRO                                  */*
*/********************************************************************/*
WTO_LIST WTO   TEXT=*-*,MF=L
WTO_LEN  EQU   *-WTO_LIST
*/********************************************************************/*
*/* DEFINE TEXT MESSAGES USED BY EXIT                                */*
*/********************************************************************/*
MESSAGES      DS OF
MSG1          DC CL100' '
             ORG MSG1
              DC C'$EXT1101I '
MSG1_JOBTYPE  DC CL3' '
              DC C' '
MSG1_JOBNAME  DC CL8' '
              DC C' SELECTED TO USE SPOOL PARTITION ON SYSID '
MSG1_SYSID    DC CL4' '
             ORG
MSG2          DC CL100'$EXT1102E $DAS COULD NOT BE FOUND ON JES2 AREA'
MSG3          DC CL100' '
             ORG MSG3
              DC C'$EXT1103I VOLUME '
MSG3_VOLUME   DC CL6' '
              DC C' ADDED TO '
MSG3_JOBTYPE  DC CL3' '
              DC C' '
MSG3_JOBNAME  DC CL8' '
              DC C' AS '
MSG3_VOLTYPE  DC CL7' '
              DC C' SPOOL VOLUME'
             ORG
MSG4          DC CL100' '
             ORG MSG4
              DC C'$EXT1104W NO SPOOL VOLUMES SELECTED TO JOB '
MSG4_JOBNAME  DC CL8' '
             ORG
MSG5          DC CL100' '
             ORG MSG5
              DC C'$EXT1105E USERID NOT FOUND FOR JOB '
MSG5_JOBNAME  DC CL8' '
             ORG
MSG6          DC CL100' '
             ORG MSG6
              DC C'$EXT1106I FENCING IS NOT ACTIVE TO SYSTEM '
MSG6_SYSID    DC CL4' '
             ORG
MSG7          DC CL100' '
             ORG MSG7
              DC C'$EXT1107E ERROR ON ACCESS JCT EXTENSION FOR JOB '
MSG7_JOBNAME  DC CL8' '
```

```
             ORG
*/*******************************************************************/*
*/* DEFINE Z/OS MAPPING MACROS                                      */*
*/*******************************************************************/*
         EJECT
         LTORG
         IEESMCA
         $MODEND
         END
```

E.3 Exit 12 program source code

The JES2 Exit 12 sample code that is used to implement the spool partitioning function to JES2 that is based on RACF profiles is shown in Example E-4.

Example E-4 JES2 Exit 12 sample

```
TITLE 'JES2 EXIT012 - SPOOL PARTITIONING'
*/*******************************************************************/*
*/* PROGRAM  - JES2X012                                             */*
*/*                                                                 */*
*/* FUNCTION - THIS EXIT WAS CODED TO WORK AS COMPLEMENT OF EXIT 11 */*
*/*           TO ADD MORE SPOOL SPACE TO A JOB                      */*
*/*******************************************************************/*
         EJECT
         PRINT GEN
*/*******************************************************************/*
*/* COPY OF $HASPGBL MACRO                                          */*
*/*******************************************************************/*
         COPY  $HASPGBL
         EJECT
*/*******************************************************************/*
*/* JES2 MACROS EXPANSION                                           */*
*/*******************************************************************/*
JES2X012 $MODULE ENVIRON=USER,                                        X
               RMODE=ANY,                                             X
               IBMJES2=SAMPLE,                                        X
               $BUFFER,              REQ BY $REQBUF, $FREEBUF         X
               $CAT,                 REQ BY HCT                       X
               $CNVWORK,             CONV. PROCESSOR PCE WORK1 AREA   X
               $DAS,                 IOT MAPPINT MACRO                X
               $DTE,                 REQ BY PCE                       X
               $DTECNV,              REQ BY DTE                       X
               $ERA,                 REQ BY DTE                       X
               $HASPEQU,             HASP EQUATES                     X
               $HCCT,                REQ BY $SAVE, $RETURN, ETC       X
               $HCT,                 REQ BY $SAVE, $RETURN, ETC       X
               $IOT,                 IOT MAPPINT MACRO                X
               $JCT,                 REQ BY CAT                       X
               $JCTX,                REQ BY CAT                       X
               $JQE,                 REQ BY HCT                       X
               $MIT,                 REQ BY MODEND                    X
               $PADDR,               REQ BY HCT                       X
               $PCE,                 REQ BY HCT                       X
               $SCAT,                REQ BY HCT                        X
               $TAB,                 REQ BY $CNVWORK                   X
               $TQE,                 REQ BY $CNVWORK                   X
               $XECB,                REQ BY DTE                        X
               $XIT,                                                   X
               CVT,                  COMMUNICATION VECTOR TABLE        X
               DEB,                  DATA EXTEND BLOCK                 X
               CNMB,                 CONVERTER MESSAGE BUFFER          X
               RPL,                  ACB REQUEST PARAMETER LIST        X
               SDWA,                 SYSTEM DIAGNOSIS WORK1ING AREA    X
               WPL                   REQ BY $$WTO
*/*******************************************************************/*
*/* JES2 EXIT 11 ENTRY POINT                                        */*
*/*******************************************************************/*
EXIT012  $ENTRY BASE=R12,CSECT=YES
         SAVE  (14,12)
         LR    R12,R15
         LM    R7,R9,0(R1)
         USING IOT,R7
         USING JCT,R8
         USING HCCT,R11
         L     R2,CCTHCT
         USING HCT,R2
*/*******************************************************************/*
*/* OBTAIN STORAGE TO VARIABLES USED BY EXIT                        */*
*/*******************************************************************/*
STORAGE_OBTAIN EQU *
         STORAGE OBTAIN,LENGTH=WORKLEN,LOC=31
         ST    R13,4(R1)
         ST    R1,8(R13)
         LR    R13,R1
         USING WORKAREA,R13
*/*******************************************************************/*
*/* VERIFY MAIN CONDITIONS TO PROCESS THE EXIT                      */*
*/*******************************************************************/*
TEST_EXIT_CALL EQU *
         XC    RETURN_CODE,RETURN_CODE
         LTR   R8,R8
         BZ    END_OF_EXIT
         TM    JCTUSERB,X'BO'
         BNO   END_OF_EXIT
         TM    JCTJOBFL,JCTBATCH
         BO    JOB_BATCH
         TM    JCTJOBFL,JCTTSUJB
         BO    JOB_TSU
         TM    JCTJOBFL,JCTSTCJB
         BO    JOB_STC
         B     END_OF_EXIT
*/*******************************************************************/*
*/* MOVE THE CORRESPONDING JOB TYPE VALUE TO RACF PROFILE           */*
*/*******************************************************************/*
JOB_BATCH EQU *
         MVC   RACF_JOB_TYPE,RACF_PROF_JOB
         B     PROCESS_EXIT
JOB_TSU  EQU *
```

```
         MVC   RACF_JOB_TYPE,RACF_PROF_TSU
         B     PROCESS_EXIT
JOB_STC  EQU *
         MVC   RACF_JOB_TYPE,RACF_PROF_STC
*/*******************************************************************/*
*/* GET REQUIRED INFORMATION FROM JCT, IOT AND HCT                  */*
*/*******************************************************************/*
PROCESS_EXIT EQU *
         MVC   #SPOOL_VOLUMES,$SPOLNUM
         MVC   #FENCE_VOLUMES,$FNCCNT
         MVC   JOB_NAME,JCTJNAME
         MVC   JOB_NUMBER,JCTJBNUM
         MVC   JOB_JOBID,JCTJOBID
         MVC   JOB_CLASS,JCTJCLAS
         PERFORM GET_SYSID,R
         PERFORM GET_USERID,R
         MVC   SPOOL_ALLOCATED_MASK,IOTSPMSK
         MVC   SPOOL_AVAILABLE_MASK,CCTVBLOB
         MVC   SPOOL_ALLOWED_MASK,0(R9)
         NC    SPOOL_ALLOWED_MASK,CCTSPLAF
         $CNTBIT FIELD=SPOOL_ALLOWED_MASK
         L     R2,CCTHCT
         STC   R1,#SPOOL_USED
*/*******************************************************************/*
*/* VERIFY IF THE FENCE PARAMETER TO SPOOL ALLOCATION IS ACTIVE     */*
*/*******************************************************************/*
         TM    CCTSTUS,CCTSMVFN
         BO    VALIDATE_EXTENSION
         GETMSG 7,WTO_MESSAGE,MESSAGES
         MVC   WTO_MESSAGE+MSG7_SYSID-MSG7(L'SYSID),SYSID
         PERFORM SEND_WTO,R
         B     END_OF_EXIT
*/*******************************************************************/*
*/* SEND MESSAGE TO INFORM THE JOB REQUESTING ADDITIONAL SPACE      */*
*/*******************************************************************/*
VALIDATE_EXTENSION EQU *
         GETMSG 1,WTO_MESSAGE,MESSAGES
         MVC   WTO_MESSAGE+MSG1_JOBNAME-MSG1(L'JOB_NAME),JOB_NAME
       MVC WTO_MESSAGE+MSG1_JOBTYPE-MSG1(L'RACF_JOB_TYPE),RACF_JOB_TYPE
         PERFORM SEND_WTO,R
         MVC   SPOOL_VOL_TYPE,VOL_ALLOWED
*/*******************************************************************/*
*/* START THE $DAS SEARCHING TO FIND ADDITIONAL VOLUMES             */*
*/*******************************************************************/*
START_DAS_SEARCH EQU *
         L     R3,CCTDAS1
         USING DAS,R3
         XC    SPOOL_VOL_SET,SPOOL_VOL_SET
         LA    R1,SPOOL_ALLOWED_MASK
         ST    R1,SPOOL_MASK_ADDRESS
         LA    R1,SPOOL_AVAILABLE_MASK
         ST    R1,SPOOL_AVAIL_ADDRESS
         LH    R4,#SPOOL_VOLUMES
         XC    BITMASK,BITMASK
         OI    BITMASK,X'80'
```

```
*/*******************************************************************/*
*/* VALIDATE IF THE JOB IS USING THE MAXIMUM OF VOLUMES ALLOWED     */*
*/*******************************************************************/*
NEW_DAS_ENTRY EQU *
         CLC   SPOOL_VOL_SET,#FENCE_VOLUMES
         BNL   END_DAS_CHAIN
         TM    DASFLAG,DASACTIV
         BNO   NEXT_DAS_ENTRY
*/*******************************************************************/*
*/* VERIFY IF THE CURRENT VOLUME IS ALREADY IN USE BY JOB         */*
*/*******************************************************************/*
         L     R1,SPOOL_MASK_ADDRESS
         XR    R15,R15
         IC    R15,BITMASK
         EX    R15,TEST_BITMASK
         BO    NEXT_DAS_ENTRY
*/*******************************************************************/*
*/* VERIFY IF THE CURRENT VOLUME HAVE SPACE TO BE USED BY JOB       */*
*/*******************************************************************/*
         L     R1,SPOOL_AVAIL_ADDRESS
         XR    R15,R15
         IC    R15,BITMASK
         EX    R15,TEST_BITMASK
         BNO   NEXT_DAS_ENTRY
*/*******************************************************************/*
*/* VERIFY IF THE CURRENT VOLUME CAN BE USED BY JOB                 */*
*/*******************************************************************/*
         MVC   SPOOL_VOLUME,DASVOLID
         PERFORM CHECK_VOLUME_ACCESS,R
         PERFORM RACF_CHECK_AUTH,R
         LTR   R15,R15
         BNZ   NEXT_DAS_ENTRY
*/*******************************************************************/*
*/* ADD THE SPOOL VOLUME AS A VOLUME ALLOWED TO BE ALLOCATED        */*
*/*******************************************************************/*
ADD_SPOOL_VOLUME EQU *
         XR    R1,R1
         IC    R1,SPOOL_VOL_SET
         LA    R1,1(R1)
         STC   R1,SPOOL_VOL_SET
         OI    FLAG,FLAG_SPOOL
*/*******************************************************************/*
*/* SET A NEW BIT ON ALLOWED BITMASK FOR THE JOB                    */*
*/*******************************************************************/*
         L     R1,SPOOL_MASK_ADDRESS
         OC    0(L'BITMASK,R1),BITMASK
*/*******************************************************************/*
*/* SEND MESSAGE TO CONSOLE WITH SPOOL VOLUME ADDED TO JOB          */*
*/*******************************************************************/*
         GETMSG 3,WTO_MESSAGE,MESSAGES
         MVC  WTO_MESSAGE+MSG3_VOLUME-MSG3(L'SPOOL_VOLUME),SPOOL_VOLUME
         MVC   WTO_MESSAGE+MSG3_JOBNAME-MSG3(L'JOB_NAME),JOB_NAME
       MVC WTO_MESSAGE+MSG3_JOBTYPE-MSG3(L'RACF_JOB_TYPE),RACF_JOB_TYPE
     MVC WTO_MESSAGE+MSG3_VOLTYPE-MSG3(L'SPOOL_VOL_TYPE),SPOOL_VOL_TYPE
         PERFORM SEND_WTO,R
```

```
*/*******************************************************************/*
*/* JUMP TO NEXT AVAILABLE DAS ENTRY ON DAS CHAIN                   */*
*/*******************************************************************/*
NEXT_DAS_ENTRY EQU *
         XR    R1,R1
         ICM   R1,15,DASTRAKQ
         LTR   R1,R1
         BZ    END_DAS_CHAIN
         LA    R3,DASSIZC(R3)
         PERFORM NEXT_BITMASK,R
         BCT   R4,NEW_DAS_ENTRY
*/*******************************************************************/*
*/* CHECK IF WAS SET ANY VOLUME TO BE USED BY JOB                 */*
*/*******************************************************************/*
END_DAS_CHAIN EQU *
         TM    FLAG,FLAG_SPOOL
         BNO   VALIDATE_OVERFLOW
         MVC   0(L'SPOOL_ALLOWED_MASK,R9),SPOOL_ALLOWED_MASK
         B     END_EXIT_08
*/*******************************************************************/*
*/* VALIDATE IF THE JOB IS CANDIDATE TO OVERFLOW THE SPOOL          */*
*/*******************************************************************/*
VALIDATE_OVERFLOW EQU *
         CLEAR RACF_PROFILE
         MVC   RACF_TYPE(L'RACF_PROF_TYPE),RACF_PROF_TYPE
         MVC   RACF_SPOOLE(L'RACF_PROF_SPOOLE),RACF_PROF_SPOOLE
         MVC   RACF_SPOOLE_SYSID(L'SYSID),SYSID
         MVI   RACF_SPOOLE_DOT,C'.'
         MVC   RACF_SPOOLE_JOBNAME(L'JOB_NAME),JOB_NAME
         PERFORM RACF_CHECK_AUTH,R
         LTR   R15,R15
         BNZ   PROCESS_OPERATOR_REQUEST
*/*******************************************************************/*
*/* SET FLAG TO SEARCH THE DAS CHAIN FOR OVERFLOW VOLUMES AVAILABLE */*
*/*******************************************************************/*
GET_OVERFLOW_VOLUMES EQU *
         TM    FLAG,FLAG_OVERFLOW
         BO    GET_DEFAULT_VOLUMES
         OI    FLAG,FLAG_OVERFLOW
         MVC   SPOOL_VOL_TYPE,VOL_OVERFLOW
         B     START_NEW_DAS_SEARCH
*/*******************************************************************/*
*/* SET FLAG TO SEARCH THE DAS CHAIN FOR OVERFLOW VOLUMES AVAILABLE */*
*/*******************************************************************/*
GET_DEFAULT_VOLUMES EQU *
         TM    FLAG,FLAG_DEFAULT
         BO    PROCESS_OPERATOR_REQUEST
         OI    FLAG,FLAG_DEFAULT
         MVC   SPOOL_VOL_TYPE,VOL_DEFAULT
         B     START_NEW_DAS_SEARCH
*/*******************************************************************/*
*/* SEND A MESSAGE INDICATING THE START OF A NEW SEARCH ON $DAS     */*
*/*******************************************************************/*
START_NEW_DAS_SEARCH EQU *
         GETMSG 9,WTO_MESSAGE,MESSAGES
```

```
       MVC WTO_MESSAGE+MSG9_JOBTYPE-MSG9(L'RACF_JOB_TYPE),RACF_JOB_TYPE
         MVC   WTO_MESSAGE+MSG9_JOBNAME-MSG9(L'JOB_NAME),JOB_NAME
     MVC WTO_MESSAGE+MSG9_VOLTYPE-MSG9(L'SPOOL_VOL_TYPE),SPOOL_VOL_TYPE
         PERFORM SEND_WTO,R
         B     START_DAS_SEARCH
*/********************************************************************/*
*/* DISPLAY MESSAGES WITH NO MORE VOLUMES AVAILABLE                  */*
*/********************************************************************/*
PROCESS_OPERATOR_REQUEST EQU *
         GETMSG 4,WTO_MESSAGE,MESSAGES
         MVC   WTO_MESSAGE+MSG4_JOBNAME-MSG4(L'JOB_NAME),JOB_NAME
         PERFORM SEND_WTO,R
*/********************************************************************/*
*/* PROCESS WTOR REQUEST AND WAIT FOR OPERATOR DECISION              */*
*/********************************************************************/*
PROCESS_WTOR EQU *
         GETMSG 8,WTOR_MESSAGE,MESSAGES
      MVC WTOR_MESSAGE+MSG8_JOBTYPE-MSG8(L'RACF_JOB_TYPE),RACF_JOB_TYPE
         MVC   WTOR_MESSAGE+MSG8_JOBNAME-MSG8(L'JOB_NAME),JOB_NAME
         PERFORM SEND_WTOR,R
*/********************************************************************/*
*/* PROCESS THE ANSWER FROM OPERATOR CONSOLE                         */*
*/********************************************************************/*
         CLI   WTOR_RESP,C'C'
         BE    SET_ALL_VOLUMES
         CLI   WTOR_RESP,C'R'
         BNE   PROCESS_WTOR
         NI    FLAG,255-(FLAG_OVERFLOW+FLAG_SPOOL+FLAG_DEFAULT)
         B     START_DAS_SEARCH
*/********************************************************************/*
*/* SEARCH RACF TO FIND A VOLUME WITH ACCESS TO BE USED              */*
*/********************************************************************/*
CHECK_VOLUME_ACCESS EQU *
         CLEAR RACF_PROFILE
         MVC   RACF_TYPE(L'RACF_PROF_TYPE),RACF_PROF_TYPE
         TM    FLAG,FLAG_OVERFLOW
         BO    CHECK_OVERFLOW_PROFILE
         TM    FLAG,FLAG_DEFAULT
         BO    CHECK_DEFAULT_PROFILE
*/********************************************************************/*
*/* CHECK IF THE VOLUME CAN BE USED BY JOB                           */*
*/********************************************************************/*
         MVC   RACF_SPOOLV(L'RACF_PROF_SPOOLV),RACF_PROF_SPOOLV
         MVC   RACF_SPOOLV_SYSID(L'SYSID),SYSID
         MVI   RACF_SPOOLV_DOT,C'.'
         MVC   RACF_SPOOLV_VOLUME(L'SPOOL_VOLUME),SPOOL_VOLUME
         BR    R10
*/********************************************************************/*
*/* VALIDATE THE OVERFLOW PROFILE TO A VOLUME                        */*
*/********************************************************************/*
CHECK_OVERFLOW_PROFILE EQU *
         MVC   RACF_SPOOLO(L'RACF_PROF_SPOOLO),RACF_PROF_SPOOLO
         MVC   RACF_SPOOLO_SYSID(L'SYSID),SYSID
         MVI   RACF_SPOOLO_DOT,C'.'
         MVC   RACF_SPOOLO_VOLUME(L'SPOOL_VOLUME),SPOOL_VOLUME
         BR    R10
*/*******************************************************************/*
*/* VALIDATE THE DEFAULT PROFILE FOR A AVAILABLE VOLUME             */*
*/*******************************************************************/*
CHECK_DEFAULT_PROFILE EQU *
         MVC   RACF_SPOOLD(L'RACF_PROF_SPOOLD),RACF_PROF_SPOOLD
         MVC   RACF_SPOOLD_SYSID(L'SYSID),SYSID
         MVI   RACF_SPOOLD_DOT,C'.'
         MVC   RACF_SPOOLD_VOLUME(L'SPOOL_VOLUME),SPOOL_VOLUME
         BR    R10
*/*******************************************************************/*
*/* SET ALL MASK BITS TO USE OVERFLOW VOLUMES                       */*
*/*******************************************************************/*
SET_ALL_VOLUMES EQU *
         MVC   0(L'SPOOL_ALLOWED_MASK,R9),FULL_MASK
         GETMSG 5,WTO_MESSAGE,MESSAGES
         MVC   WTO_MESSAGE+MSG5_JOBNAME-MSG5(L'JOB_NAME),JOB_NAME
         PERFORM SEND_WTO,R
         B     END_EXIT_08
*/*******************************************************************/*
*/* POINT TO NEXT BITMAKS TO BE USED FOR SPOOL VOLUME               */*
*/*******************************************************************/*
NEXT_BITMASK EQU *
         TM    BITMASK,X'01'
         BO    SHIFT_SPOOL_MASK
         XR    R1,R1
         IC    R1,BITMASK
         SRL   R1,1
         STC   R1,BITMASK
         BR    R10
*/*******************************************************************/*
*/* WALK THRU SPOOL_ALLOWED_MASK TO NEXT 32 BITS                    */*
*/*******************************************************************/*
SHIFT_SPOOL_MASK EQU *
         L     R1,SPOOL_MASK_ADDRESS
         LA    R1,L'BITMASK(R1)
         ST    R1,SPOOL_MASK_ADDRESS
         L     R1,SPOOL_AVAIL_ADDRESS
         LA    R1,L'BITMASK(R1)
         ST    R1,SPOOL_AVAIL_ADDRESS
         XC    BITMASK,BITMASK
         OI    BITMASK,X'80'
         BR    R10
*/*******************************************************************/*
*/* ROUTINE TO GET SYSID FROM SYSTEM                                */*
*/*******************************************************************/*
GET_SYSID EQU *
         L     R1,CVTPTR
         L     R1,CVTSMCA-CVTMAP(R1)
         USING SMCABASE,R1
         MVC   SYSID,SMCASID
         BR    10
*/*******************************************************************/*
*/* VERIFY IF EXISTS USER PARAMETER ON JOB CARD OR GET A USER       */*
*/*******************************************************************/*
```

```
GET_USERID EQU *
         CLEAR JOB_USERID
         CLI   JCTNOUSR,X'00'
         BNE   USERID_FOUND
         GETMSG 6,WTO_MESSAGE,MESSAGES
         MVC   WTO_MESSAGE+MSG6_JOBNAME-MSG6(L'JOB_NAME),JOB_NAME
         PERFORM SEND_WTO,R
         B     END_OF_EXIT
*/*******************************************************************/*
*/* GET THE USERID FROM JCT SUBMITTING USER                         */*
*/*******************************************************************/*
USERID_FOUND EQU *
         OI    FLAG,FLAG_USERID
         MVC   JOB_USERID(8),JCTNOUSR
         TM    JCTFLAG1,JCT1UNDF
         BOR   R10
         CLC   JCTJUSID,=8X'00'
         BER   R10
         CLC   JCTJUSID,JCTNOUSR
         BER   R10
         MVC   JOB_USERID(8),JCTJUSID
         BR    R10
*/*******************************************************************/*
*/* SUBROUTINE TO REQUEST RACF ACCESS VALIDATION                    */*
*/*******************************************************************/*
RACF_CHECK_AUTH EQU *
         STM   R3,R4,SAVE34
         MVC   RACFT(RACLEN),RAC_LIST
         LA    R3,RACF_PROFILE
         LA    R4,RACF_CLASS_FACILITY
         RACROUTE REQUEST=AUTH,                                        X
               WORKA=RACWORK,                                          X
               ENTITY=((3)),                                           X
               CLASS=((4)),                                            X
               ATTR=READ,                                              X
               GENERIC=ASIS,                                           X
               USERID=JOB_USERID,                                      X
               RELEASE=7790,                                           X
               LOG=NONE,                                               X
               MF=(E,RACFT)
         LM    R3,R4,SAVE34
         BR    10
*/*******************************************************************/*
*/* SUBROUTINE TO SEND A MESSAGE TO CONSOLE                         */*
*/*******************************************************************/*
SEND_WTO EQU  *
         MVC   WTO_MSGL,=AL2(L'WTO_MESSAGE)
         MVC   WTO_EXEC,WTO_LIST
         $$WTO WTO_EXEC,TEXT=WTO_MSG
         BR    R10
*/*******************************************************************/*
*/* SUBROUTINE TO SEND A WTOR REQUEST TO CONSOLE                    */*
*/*******************************************************************/*
SEND_WTOR EQU  *
         XC    WTOR_ECB,WTOR_ECB
         MVC   WTOR_MSGL,=AL2(L'WTOR_MESSAGE)
         MVC   WTOR_EXEC,WTOR_LIST
         WTOR  TEXT=(WTOR_MSG,WTOR_RESP,WTOR_RLEN,WTOR_ECB),           X
               MF=(E,WTOR_EXEC,EXTENDED)
         WAIT  ECB=WTOR_ECB
         BR    R10
*/*****************************************************************/*
*/* END OF EXIT - RELEASE ACQUIRED STORAGE AND RETURN TO CALLER     */*
*/*****************************************************************/*
END_EXIT_08 EQU *
         MVC   RETURN_CODE,FULL_8
END_OF_EXIT EQU *
         L     R15,RETURN_CODE
         LR    R1,R13
         L     R13,4(R13)
         ST    R15,16(R13)
         STORAGE RELEASE,LENGTH=WORKLEN,ADDR=(1)
         LM    R14,R12,12(R13)
         BR    R14
*/*****************************************************************/*
*/* INSTRUCTION AREA USED TO EXECUTE                                */*
*/*****************************************************************/*
TEST_BITMASK TM 0(R1),X'00'
*/*****************************************************************/*
*/* WORKAREA OBTAINED BY EXIT                                       */*
*/*****************************************************************/*
WORKAREA       DSECT
SAVEAREA       DS 18F
SAVE34         DS 2F
DOUBLE         DS D
RETURN_CODE    DS F
BITMASK        DS X
#SPOOL_VOLUMES DS H
#SPOOL_USED    DS F
#FENCE_VOLUMES DS X
JOB_NUMBER     DS F
JOB_NAME       DS CL8
JOB_USERID     DS CL8
JOB_CLASS      DS CL1
JOB_JOBID      DS CL8
SYSID          DS CL4
SPOOL_MASK_ADDRESS   DS F
SPOOL_AVAIL_ADDRESS  DS F
SPOOL_ALLOCATED_MASK DS 8F
SPOOL_AVAILABLE_MASK DS 8F
SPOOL_ALLOWED_MASK   DS 8F
SPOOL_VOLUME         DS CL6
SPOOL_VOL_SET        DS X
SPOOL_VOL_TYPE       DS CL7
FLAG           DS X
FLAG_CLASS    EQU X'01'
FLAG_SPOOL    EQU X'02'
FLAG_USERID   EQU X'04'
FLAG_JOBNAME  EQU X'08'
FLAG_CANCEL   EQU X'10'
```

```
FLAG_OVERFLOW EQU X'20'
FLAG_DEFAULT  EQU X'40'
*/********************************************************************/*
*/* DEFINE AREA TO MAP RACF PROFILES TO BE USED                      */*
*/********************************************************************/*
RACF_JOB_TYPE     DS CL3
RACF_PROFILE      DS CL44
         ORG RACF_PROFILE
RACF_TYPE         DS CL(L'RACF_PROF_TYPE)
RACF_TYPE_VAR     DS OC
         ORG
         ORG RACF_TYPE_VAR
RACF_SPOOLC       DS CL(L'RACF_PROF_SPOOLC)
RACF_SPOOLC_SYSID DS CL4
RACF_SPOOLC_DOT   DS CL1
RACF_SPOOLC_CLASS DS CL1
         ORG
         ORG RACF_TYPE_VAR
RACF_SPOOLV       DS CL(L'RACF_PROF_SPOOLV)
RACF_SPOOLV_SYSID DS CL4
RACF_SPOOLV_DOT   DS CL1
RACF_SPOOLV_VOLUME DS CL6
         ORG
         ORG RACF_TYPE_VAR
RACF_SPOOLD       DS CL(L'RACF_PROF_SPOOLD)
RACF_SPOOLD_SYSID DS CL4
RACF_SPOOLD_DOT   DS CL1
RACF_SPOOLD_VOLUME DS CL6
         ORG
         ORG RACF_TYPE_VAR
RACF_SPOOLO       DS CL(L'RACF_PROF_SPOOLO)
RACF_SPOOLO_SYSID DS CL4
RACF_SPOOLO_DOT   DS CL1
RACF_SPOOLO_VOLUME DS CL6
         ORG
         ORG RACF_TYPE_VAR
RACF_SPOOLE       DS CL(L'RACF_PROF_SPOOLE)
RACF_SPOOLE_SYSID DS CL4
RACF_SPOOLE_DOT   DS CL1
RACF_SPOOLE_JOBNAME DS CL8
         ORG
*/********************************************************************/*
*/* DEFINE CLASS TO USE WITH RACF AND LIST FORM OF MACRO RACROUTE    */*
*/********************************************************************/*
         DS    OF
RACWORK  DS    CL512                   WORK AREA DO RACF
         DS    OF
RACFT    RACROUTE REQUEST=AUTH,                                        X
               WORKA=*-*,                                              X
               CLASS='FACILITY',                                       X
               ATTR=READ,                                              X
               RELEASE=7790,                                           X
               MF=L
*/********************************************************************/*
*/* DEFINE WORK AREA TO BE USED BY WTO MACRO                         */*
```

```
*/********************************************************************/*
WTO_EXEC DS    CL(WTO_LEN)
WTO_MSG  DS    0H
WTO_MSGL DS    AL2
WTO_MESSAGE DS CL100
*/********************************************************************/*
*/* DEFINE WORK AREA TO BE USED BY WTO MACRO                         */*
*/********************************************************************/*
WTOR_ECB  DS    F
WTOR_RLEN EQU L'WTOR_RESP
WTOR_RESP DS    CL1
WTOR_EXEC DS    CL(WTOR_LEN)
WTOR_MSG  DS    0H
WTOR_MSGL DS    AL2
WTOR_MESSAGE DS CL100
*/********************************************************************/*
*/* DEFINE AREA USED BY PERFORM PROCESS                              */*
*/********************************************************************/*
         PERFORM GENERATE
WORKLEN  EQU   *-WORKAREA
         EJECT
*/********************************************************************/*
*/* USED CONSTANTS BY EXIT                                           */*
*/********************************************************************/*
EXIT012 CSECT
EXIT012 AMODE 31
EXIT012 RMODE ANY
FULL_0         DC F'0'
FULL_4         DC F'4'
FULL_8         DC F'8'
FULL_32        DC F'32'
FULL_MASK      DC 32X'FF'
VOL_ALLOWED    DC C'ALLOWED'
VOL_DEFAULT    DC C'DEFAULT'
VOL_OVERFLOW   DC C'OVRFLOW'
*/********************************************************************/*
*/* DEFINE CONSTANTS TO BE USED FOR RACF PROCESSING                  */*
*/********************************************************************/*
RACF_CLASS_FACILITY DC AL1(L'FACILITY)
FACILITY            DC C'FACILITY'
RACF_PROF_TYPE    DC  C'$JES2.SPART.'
RACF_PROF_JOB     DC  C'JOB'
RACF_PROF_TSU     DC  C'TSU'
RACF_PROF_STC     DC  C'STC'
RACF_PROF_SPOOLC  DC  C'CLS.'
RACF_PROF_SPOOLV  DC  C'VOL.'
RACF_PROF_SPOOLO  DC  C'OVRFL.'
RACF_PROF_SPOOLD  DC  C'DFLT.'
RACF_PROF_SPOOLE  DC  C'EXT.'
         DS    0F
RAC_LIST RACROUTE REQUEST=AUTH,                                        X
               WORKA=*-*,                                              X
               CLASS='FACILITY',                                       X
               ATTR=READ,                                              X
               RELEASE=7790,                                           X
               MF=L
RACLEN   EQU   (*-RAC_LIST)
*/********************************************************************/*
*/* DEFINE LIST FORM TO $$WTO MACRO                                  */*
*/********************************************************************/*
WTO_LIST WTO   TEXT=*-*,MF=L
WTO_LEN  EQU   *-WTO_LIST
*/********************************************************************/*
*/* DEFINE LIST FORM TO WTOR MACRO                                   */*
*/********************************************************************/*
WTOR_LIST WTOR  TEXT=(WTOR_MSG,*-*,WTOR_RLEN,*-*),MF=L
WTOR_LEN  EQU   *-WTOR_LIST
*/********************************************************************/*
*/* DEFINE TEXT MESSAGES USED BY EXIT                                */*
*/********************************************************************/*
MESSAGES      DS OF
MSG1          DC CL100' '
             ORG MSG1
              DC C'$EXT1201I '
MSG1_JOBTYPE  DC CL3' '
              DC C' '
MSG1_JOBNAME  DC CL8' '
              DC C' REQUESTED TO USE ADDITIONAL SPOOL VOLUME'
             ORG
MSG2          DC CL100'$EXT1202E $DAS COULD NOT BE FOUND ON JES2 AREA'
MSG3          DC CL100' '
             ORG MSG3
              DC C'$EXT1203I VOLUME '
MSG3_VOLUME   DC CL6' '
              DC C' ADDED TO '
MSG3_JOBTYPE  DC CL3' '
              DC C' '
MSG3_JOBNAME  DC CL8' '
              DC C' AS '
MSG3_VOLTYPE  DC CL7' '
              DC C' SPOOL VOLUME'
             ORG
MSG4          DC CL100' '
             ORG MSG4
              DC C'$EXT1204W SPOOL PARTITIONING FOR JOB '
MSG4_JOBNAME  DC CL8' '
              DC C' IS FULL. NO OVERFLOW IS POSSIBLE'
             ORG
MSG5          DC CL100' '
             ORG MSG5
              DC C'$EXT1205W SPOOL OVERFLOWED TO JOB '
MSG5_JOBNAME  DC CL8' '
             ORG
MSG6          DC CL100' '
             ORG MSG6
              DC C'$EXT1206E THE USERID FOR '
MSG6_JOBNAME  DC CL8' '
              DC C' COULD NOT BE FOUND'
             ORG
MSG7          DC CL100' '
```

```
             ORG MSG7
              DC C'$EXT1207I FENCING IS NOT ACTIVE TO SYSTEM '
MSG7_SYSID    DC CL4' '
             ORG
MSG8          DC CL100' '
             ORG MSG8
              DC C'$EXT1208I REPLY "R" TO RETRY OR "C" TO CANCEL '
              DC C'THE SPOOL PARTITIONING FOR '
MSG8_JOBTYPE  DC CL3' '
              DC C' '
MSG8_JOBNAME  DC CL8' '
             ORG
MSG9          DC CL100' '
             ORG MSG9
              DC C'$EXT1209I NO SPOOL VOLUMES AVAILABLE FOR '
MSG9_JOBTYPE  DC CL3' '
              DC C' '
MSG9_JOBNAME  DC CL8' '
              DC C'. SEARCHING '
MSG9_VOLTYPE  DC CL7' '
              DC C' VOLUMES'
             ORG
*/*******************************************************************/*
*/* DEFINE Z/OS MAPPING MACROS                                      */*
*/*******************************************************************/*
         EJECT
         LTORG
         IEESMCA
         $MODEND
         END
```

E.4 Other code used by exits

The CLEAR macro that is used by both JES2 exits to clear used work areas is shown in Example E-5.

Example E-5 CLEAR macro that is used by JES2 exits samples

```
MACRO
&LABEL   CLEAR &FIELD
         LCLA  &A
&LABEL   MVI   &FIELD,C' '
         MVC   &FIELD+1(L'&FIELD-1),&FIELD
         MEND
```

The Example 6-46 shows the GETMSG sample macro used by JES2 exits to get messages from message pool.

Example 6-46 Sample of GETMSG macro

```
MACRO
&NOME    GETMSG &MSG,&AREA,&CSECT
         LCLA  &A
&NOME    LA    15,&MSG
         BCTR  15,0
         MH    15,=AL2(L'&AREA)
         AIF   ('&CSECT'(1,1) EQ '(').REGOK
         A     15,=A(&CSECT)
         AGO   .MOVE
.REGOK   ANOP
&REG     SETC  '&CSECT'(2,1)
&REGNO   SETA  &REG
         AR    15,&REGNO
.MOVE    ANOP
         MVC   &AREA+0(L'&AREA),0(15)
.EXIT    MEND
```

F

Alternative conversion programs

This appendix describes non-IBM software products than can assist you in the conversion of your JES3 JCL and JECL to the JES2 equivalent. These products provide functions that might allow you to run JES3 JECL and JCL unchanged on a JES2 system.

This appendix includes the following topics:

z/OSEM

Trident Services' z/Operating System Environment Manager (z/OSEM) offers a methodology and ISPF interface that provides dynamic controls. In addition to the dynamic controls that are offered, it facilitates JES3-to-JES2 migrations by providing a migration path from JES3 to JES2 by including most of the workload routing functionality of JES3.

It also can reinterpret JOB JCL if z/OSEM Job Classing is used to change a JOB's CLASS to obtain JES2 default values, such as TIME.

z/OSEM also provides support for the JES2 parameter in the initialization deck **`JOBDEF INTERPRET=JES|INIT`**, JES2 Exit 59, and JES2 Exit 60.

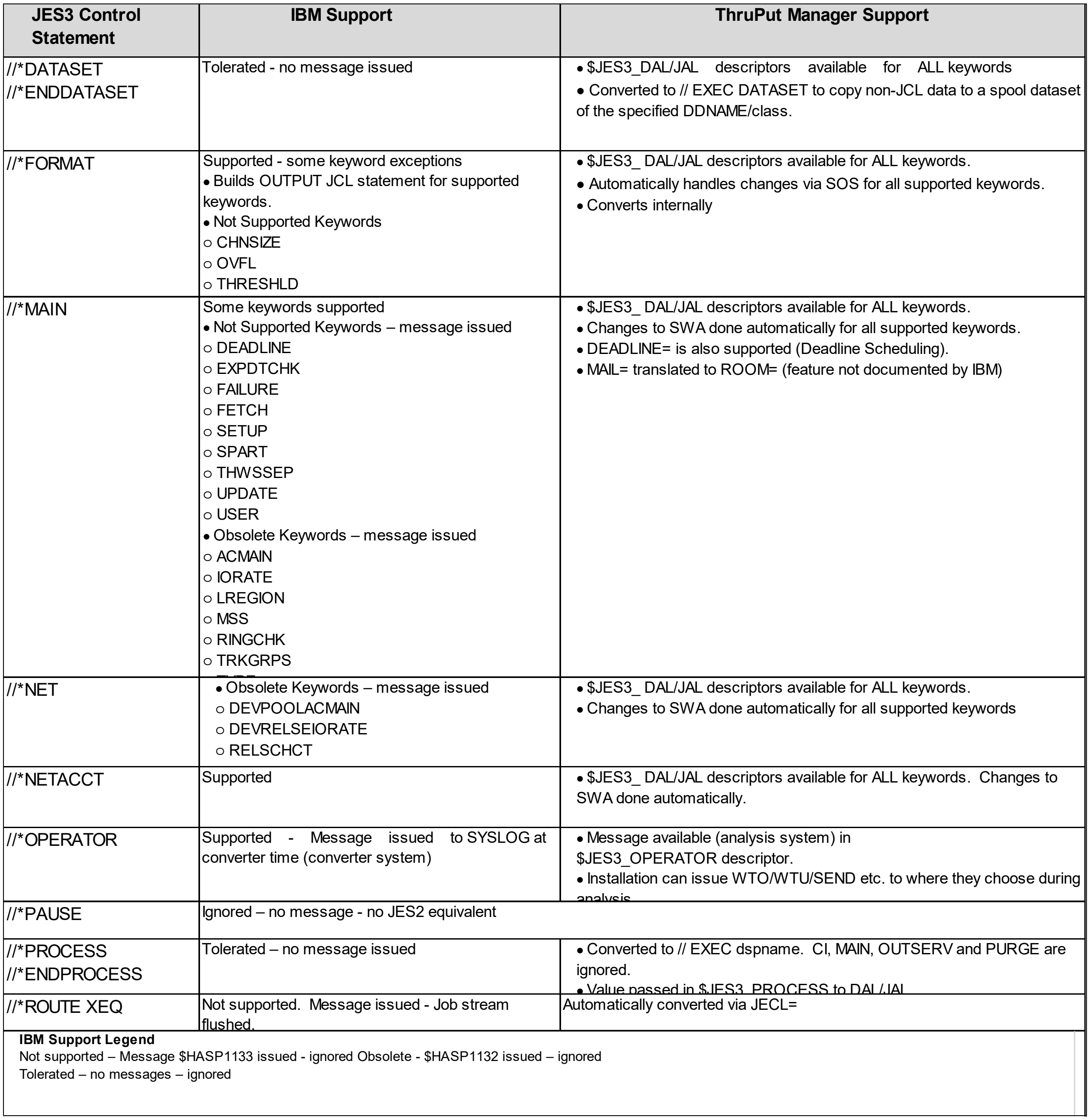
For more information, see this website.

ThruPut Manager Automation Edition

Compuware's ThruPut Manager Automation Edition does not run with JES3. However, it can be of invaluable assistance to an installation that is converting from JES3 to JES2 by providing equivalent functions to some of the JES3 capabilities that are otherwise lost or pose a significant challenge to such a conversion.

The support that is offered by IBM as of this writing and ThruPut Manager Automation Edition is compared in Figure F-1 on page 265.

For more information about ThruPut Manager Automation Edition, see this website.



JES3 Control Statement	IBM Support	ThruPut Manager Support
//*DATASET //*ENDDATASET	Tolerated - no message issued	• $JES3_DAL/JAL descriptors available for ALL keywords • Converted to // EXEC DATASET to copy non-JCL data to a spool dataset of the specified DDNAME/class.
//*FORMAT	Supported - some keyword exceptions • Builds OUTPUT JCL statement for supported keywords. • Not Supported Keywords ○ CHNSIZE ○ OVFL ○ THRESHLD	• $JES3_ DAL/JAL descriptors available for ALL keywords. • Automatically handles changes via SOS for all supported keywords. • Converts internally
//*MAIN	Some keywords supported • Not Supported Keywords – message issued ○ DEADLINE ○ EXPDTCHK ○ FAILURE ○ FETCH ○ SETUP ○ SPART ○ THWSSEP ○ UPDATE ○ USER • Obsolete Keywords – message issued ○ ACMAIN ○ IORATE ○ LREGION ○ MSS ○ RINGCHK ○ TRKGRPS TYPE	• $JES3_ DAL/JAL descriptors available for ALL keywords. • Changes to SWA done automatically for all supported keywords. • DEADLINE= is also supported (Deadline Scheduling). • MAIL= translated to ROOM= (feature not documented by IBM)
//*NET	• Obsolete Keywords – message issued ○ DEVPOOLACMAIN ○ DEVRELSEIORATE ○ RELSCHCT	• $JES3_ DAL/JAL descriptors available for ALL keywords. • Changes to SWA done automatically for all supported keywords
//*NETACCT	Supported	• $JES3_ DAL/JAL descriptors available for ALL keywords. Changes to SWA done automatically.
//*OPERATOR	Supported - Message issued to SYSLOG at converter time (converter system)	• Message available (analysis system) in $JES3_OPERATOR descriptor. • Installation can issue WTO/WTU/SEND etc. to where they choose during analysis
//*PAUSE	Ignored – no message - no JES2 equivalent	
//*PROCESS //*ENDPROCESS	Tolerated – no message issued	• Converted to // EXEC dspname. CI, MAIN, OUTSERV and PURGE are ignored. • Value passed in $JES3_PROCESS to DAL/JAL
//*ROUTE XEQ	Not supported. Message issued - Job stream flushed.	Automatically converted via JECL=

IBM Support Legend
Not supported – Message $HASP1133 issued - ignored Obsolete - $HASP1132 issued – ignored
Tolerated – no messages – ignored

Figure F-1 JES3 control statement support comparison

Note: If z/OS JES2 support for JES3 JECL statements is enabled (**$TINPUTDEF,JES3JECL=PROCESS**), ThruPut Manager Automation Edition leaves ALL JES3 JECL to be processed by IBM.

If ThruPut Manager Automation Edition JES3 support is enabled, all jobs with JES3 JECL statements receive the following message:

```
DTM1118I JES3 JECL Ignored due to 'INPUTDEF JES3JECL=PROCESS' in SYSMSGS.
```

G

Additional material

This book refers to additional material that can be downloaded from the internet, as described in the following sections.

Locating the web material

The web material that is associated with this book is available in softcopy on the internet from the IBM Redbooks web server:

`ftp://www.redbooks.ibm.com/redbooks/SG248427/Assembler.zip`

Alternatively, you can go to the IBM Redbooks website:

ibm.com/redbooks

Search for SG24-8427, select the title, and then, click **Additional materials** to open the directory that corresponds with the IBM Redbooks form number, SG24-8427.

Using the web material

The additional web material that accompanies this book includes the following file:

File name	*Description*
Assembler.zip	Zipped Assembler Code Samples

Downloading and extracting the web material

Create a subdirectory (folder) on your workstation, and extract the contents of the web material `.zip` file into this folder.

Related publication

The publication that is listed in this section is considered particularly suitable for a more detailed discussion of the topics that are covered in this book.

IBM Redbooks

The IBM Redbooks publication *JES3 to JES2 Migration Considerations*, SG24-8083, provides more information about the topics in this document. Note that this publication might be available in softcopy only.

You can search for, view, download or order this document and other Redbooks, Redpapers, Web Docs, draft, and other materials, at the following website:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

A Guide to JES3 to JES2 Migration

SG24-8427-01

ISBN 0738458201

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

SG24-8427-01

ISBN 0738458201

Printed in U.S.A.