Redbooks

ibm.com/redbooks

IBM z/OS V2R2: Operations

Keith Winnard

Natalia Barros

Jose Gilberto Biondo Jr

Wilson de Figueiredo

Paul Robert Hering

Jaqueline Mourão

Ewerton Waki

z Systems

IBM

Redbooks

International Technical Support Organization

IBM z/OS V2R2: Operations

December 2015

SG24-8305-00

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

First Edition (December 2015)

This edition applies to Version 2 Release 2 of z/OS (product number 5650-ZOS).

Contents

Notices

This information was developed for products and services offered in the U.S.A.

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, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: 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 states 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 supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

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 the names and addresses used by an actual business enterprise 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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

DB2®	MVS™	RACF®
Domino®	OS/390®	Redbooks®
GDPS®	Parallel Sysplex®	Redbooks (logo)  ®
IBM®	PrintWay™	z/OS®

The following terms are trademarks of other companies:

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

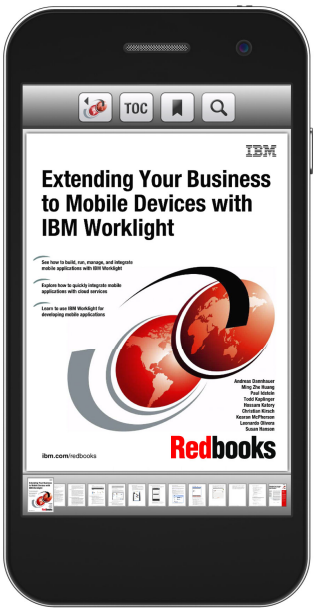
Other company, product, or service names may be trademarks or service marks of others.

Find and read thousands of IBM Redbooks publications

- Search, bookmark, save and organize favorites
- Get up-to-the-minute Redbooks news and announcements
- Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App

iOS **Download Now** Android



Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!

ibm.com/Redbooks
About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redbooks® publication helps you to become familiar with the technical changes that were introduced into the Operations areas with IBM z/OS® V2R2.

This book includes the following chapters:

- Chapter 1, “SMFLIMxx parmlib member” on page 1 describes how to set up job cancel decisions and region and MEMLIMIT values, based on the job and its environment.
- Chapter 2, “REGIONX” on page 5 describes the new REGIONX JCL keyword that supports two storage specifications and with which the user can specify values for their below-the-line storage and above-the-line storage needs.
- Chapter 3, “Enhancements in opening data sets” on page 9 describes the ability to have open up to 2,000,000 data sets at a time by address spaces while reducing the allocation time for data sets.
- Chapter 4, “Dynamic APF SMF Record” on page 11 describes the new SMF record that is used to track dynamic APF changes.
- Chapter 5, “Initial program load device number and volume” on page 13 describes the new message that is used to identify the initial program load (IPL) device and volume.
- Chapter 6, “Infoprint enhancements” on page 15 describes the enhancements that were made to Infoprint in z/OS V2R2.
- Chapter 7, “GRS EQDQ Monitor” on page 19 describes the monitor capture diagnostics for problem determination while minimizing the effect to system performance.
- Chapter 8, “BCPii System Management Facilities recording” on page 23 describes the record changes to hardware.
- Chapter 9, “Provisioning based on CPU consumption” on page 25 describes the provisioning that is dynamically based on CPU consumption.
- Chapter 10, “System REXX enhancements” on page 31 describes the new functions and interfaces of System Rexx.

This book is one of a series of IBM Redbooks publications that take a modular approach to providing information about the updates that are included with z/OS V2R2. This approach has the following goals:

- Provide modular content
- Group the technical changes into a topic
- Provide a more streamlined way of finding relevant information that is based on the topic

We hope you find this approach useful and we welcome your feedback.

Authors

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

Keith Winnard is the z/OS Project Leader at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and is keen to engage with customers to understand what they want from IBM Redbooks Publications. Before joining the ITSO in 2014, Keith worked for clients and Business Partners in the UK and Europe in various technical and account management roles. He is experienced with blending and integrating new technologies into the traditional landscape of mainframes.

Natalia Barros is a Systems Programmer at Banco do Brasil, a government bank in Brazil. She has seven years of mainframe experience and holds a degree in Computer Science. Her areas of expertise include storage architecture, mainframe hardware configuration, Data Facility Storage Management Subsystem (DFSMS), and remote copy solutions.

Jose Gilberto Biondo Jr is an IT Specialist in Integrated Technology Delivery, ServerSystems Operations/Storage Management in IBM Brazil. He has seven years of experience inz/OS, working with storage management since 2007. Jose works mainly with IBM storage products (DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm), but he also works with OEM software products. Jose's areas of expertise include installing and maintaining storage products, and process automation.

Wilson de Figueiredo is z/OS System Programmer. He manages the operations support team at Banco do Brasil, a government bank in Brazil. He has more than 11 years of experience in mainframe systems. He holds a system analysis, internet consulting, and busineSs administration degrees. His areas of expertise include IBM Parallel Sysplex®, z/OS security, and z/OS availability.

Paul Robert Hering is an IT Specialist at the ITS Technical Support Center, Mainz, Germany. He provides support to clients with z/OS and z/OS UNIX related questions and issues. He has participated in several ITSO residencies since 1988, writing about UNIX related topics. Before supporting IBM OS/390® and z/OS, Robert worked for many years with the IBM VM operating system and its variations (VM/370, VM/HPO, VM/XA, and VM/ESA).

Jaqueline Mourao is a Senior Systems Programmer at Banco do Brasil, a government bank in Brazil. She has 13 years of mainframe experience and holds a degree in Information Systems. Her areas of expertise include z/OS installation and maintenance, IBM Parallel Sysplex, and security.

Ewerton Waki is an IT Analyst at Banco do Brasil, a government bank in Brazil. He has four years of experience in mainframe systems, including IBM GDPS®, DFSMS, high-end storage systems, and remote copy solutions. He holds a Bachelor in Information Systems from Universidade Estadual de São Paulo. His area of expertise includes DFSMS and storage performance for mainframe environments.

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

Bob Haimowitz (Development Support Team [DST], Poughkeepsie Center) for setting up and maintaining the systems, and providing valuable advice, guidance, and assistance throughout the creation of this IBM Redbooks publication.

Rich Conway (DST, Poughkeepsie Center) for setting up and maintaining the systems, and providing valuable advice, guidance, and assistance throughout the creation of this IBM Redbooks publication.

Peter Bertolozzi (Systems Management specialist, IBM Redbooks residency support, Poughkeepsie Center) for setting up and maintaining the environments within syslab in which residents worked.

John Gierloff (Operations, Poughkeepsie Center) for residency set up and support.

Don Brennan (DST, Poughkeepsie Center) for setting up and maintaining the systems hardware used in the creation of this IBM Redbooks publication.

Ella Buslovich (Graphics specialist, location) for providing guidance and specialist graphics for this IBM Redbooks publication.

Ann Lund (ITSO Administration, Poughkeepsie Center) for administrative support to enable the residences publication.

Cheryl Gera (ITSO Administration, Poughkeepsie Center) for managing the business operations for this IBM Redbooks publication.

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

1

SMFLIMxx parmlib member

This chapter describes the new parmlib member SMFLIMxx with which you can set up job cancel decisions and region and MEMLIMIT values that are based on the job and its environment.

1.1 SMFLIMxx overview

To control memory usage by JOBs, systems administrators might decide to implement IEFUSI exit to assign REGION and MEMLIMIT values to jobs and programs that are based on specific information. This configuration ensures that proper amounts of working storage are allocated for user program and system use within an address space.

Although IEFUSI provides the means to control storage, it does not provide an alternative to specify the amount of storage to reserve for system use. It also might require high overhead for changing code to update rules about execution.

z/OS V2R2 introduces a new SMFLIMxx parmlib member with which system administrators can set up job cancel decisions and region and MEMLIMIT values that are based on the job and its environment.

1.1.1 Activation

The new SMFLIMxx parmlib member defines job storage usage and cancel decisions that are often performed by IEFUSI exit. This configuration reduces the number of required updates to IEFUSI exit and the time that is required to perform operation changes.

The following commands can be used to activate and test SMFLIMxx member settings:

- `SMFLIM=(xx)`

  This command is used to activate the SMFLIMxx member at initial program load (IPL) time, where xx is the last two characters.

- `SET SMFLIM=(xx)`

  This command is the operator command that is used command to activate or refresh the SMFLIMxx member without an IPL.

- `SET SMFLIM=(xx,C)`

  This command is the operator command that is used to perform a syntax check, where xx is the last two characters of the SMFLIMxx member. The C keyword performs the syntax check. This command does *not* activate new SMFLIM rules.

1.1.2 Filters

Within SMFLIMxx member, the following filter keywords can be used to define rules (each keyword has 1 - 8 values):

- `JOBNAME`
- `JOBACCT`
- `STEPNAME`
- `STEPACCT`
- `PGMNAME`
- `USER`
- `JOBCLASS`
- `SUBSYS`
- `SYSNAME`

You can use " * " and " ? " as wildcards for all filter keywords and " % " for `SETPACCT/JOBACCT` that indicates there are no matches on the remaining fields.

The following keywords define the storage that is available for use below and above the line. The use of these keywords provides a granularity in defining storage:

- REGIONABOVE

  This keyword is used to indicate the amount of storage that is available for processing by the JOB *above* the line. Its values are 1 M - 2 G.

- REGIONBELOW

  This keyword indicates the amount of storage that is available for processing by the JOB *below* the line. Its values are 1 K - 16 M.

- SYSRESABOVE

  This keyword reserves a specified amount of extended storage for system use (for functions, such as task termination and SMF recording). Its values are 1 M - 2 G.

- SYSRESBELOW

  This keyword reserves a specified amount of non-extended storage for system use. Its values are 1 K - 16 M.

- MEMLIMIT

  This keyword specifies the limit on the total size of usable virtual storage that is above the bar in a single address space. Its value can be defined in M, G, T, and P, up to 16384 P.

- EXECUTE

  This keyword defines whether the JOB can run. Possible values are YES, CANCEL, CANCELFROMIEFUSI (use decision from IEFUSI), and NOCHANGE (default).

All statements within MEMLIMxx member must start with REGION keyword, followed by set of filter keywords that indicate to which jobs or jobsteps to apply the settings, and one or more settings to apply. Example 1-1 shows a simple implementation to reserve 4 M of storage below the bar and 1 G of storage above the bar for IBM DB2® JOBs.

Example 1-1 Sample SMFLIM rule

```
REGION JOBNAME(DB2*)
REGIONBELOW(4M) REGIONABOVE(1G)
```

If there is more than one rule that applies to a specific job or step, a mix of the settings is used to define the rule with later keywords overriding previous values. Example 1-2 shows a simple configuration in which Storage JOBs that are running ADRDSSU programs have 6 M of storage below the line and a MEMLIMIT of 4 G.

Example 1-2 Sample SMFLIM override match value

```
REGION JOBNAME(STOR*)
REGIONBELOW(6M) MEMLIMIT(1G)
REGION PGMNAME(ADRDSSU)
MEMLIMIT(4G)
```

Note: Although the first rule sets MEMLIMIT for 1 G on STOR jobs, the second matching rule overrides this value.

Consideration must be taken when coding JOBACCT or STEPACCT filters to your rules because they can be omitted or coded in different ways, as shown in the following examples:

- To match a job that does not specify accounting data at all, use the JOBACCT() or JOBACCT(,<other job acct strings>) forms.

- To match when accounting keyword is present but data is null, use JOBACCT(()) or JOBACCT((),<other job acct strings>).
- Use JOBACCT(,()) to cover both options.

Example 1-3 shows a sample implementation of JOBACCTE with the % wildcard. As shown in this example, any JOBs that have DB2 as first account code value are matched and at least two values in account code field.

Example 1-3 Sample SMFLIM rule

```
REGION JOBACCT((DB2,%))
REGIONBELOW(4M) REGIONABOVE(1G) EXECUTE(YES)
```

When SMFLIMxx applies a limit, it writes a multi-line message to the job log, which indicates what changed and where the rule for it originated, as shown in Example 1-4.

Example 1-4 Sample IEF043I message

```
IEF043I Actions taken by SMFLIMxx parmlib policy for USI00011 USI00011
        Step region below changed from 00000M to 00001M by policy - SMFLINHA 0005
        Step region above changed from 00000M to 00001G by policy - SMFLINHA 0006
```

1.1.3 IEFUSI considerations

IEFUSI can be updated to override SMFLIMxx in all or some cases. A new parameter bit (word 1, bit 4, pointed to from word 5 of the parameter list) bypasses SMFLIMxx.

IEFUSI can be updated to remove all REGION/MEMLIMIT decisions. Be sure to leave in any processing that is required for other exits, such as values that are passed to other IEFUTL or IEFUSO.

2

REGIONX

This chapter describes the new REGIONX keyword on the JOB and EXEC JCL statements. It also describes the differences and similarities between the REGIONX and REGION keywords.

This chapter includes the following topics:

2.1 REGIONX overview

To run their jobs and applications in controlled environments, users want more control over their below-the-line and above-the-line storage requests. Although the old REGION specification can be used to control memory usage, users cannot specify values for below- and above-the-line storage. Because REGION uses its own rules for defining below- and above-the-line values, a user can have more below-the-line storage than they want and less above-the-line storage than they need.

The new REGIONX JCL keyword supports two storage specifications and with which the user can explicitly specify values for their below-the-line storage and above-the-line storage needs. This specification allows an installation to reduce the amount of below-the-line storage that must be available to process jobs.

2.2 Use of REGIONX

You can code the REGIONX keyword on JOB and EXEC JCL statements, but it is mutually exclusive with the REGION keyword. Therefore, if REGIONX is coded in your JOB card, none of the JOB steps can have REGION set, and vice versa.

Note: Although you can have REGION and REGIONX in different steps of your JOB, we suggest you that you avoid this configuration if you do not code it in the JOB card because updates to your JCL can cause the mutually exclusive JCL error.

To define the amount of required storage, you can code below-the-line, above-the-line, or both. Table 2-1 lists possible REGIONX configurations.

Table 2-1 Possible REGIONX configurations

REGIONX configuration	Storage allocation
REGIONX=	Both values default.
REGIONX=1M	1 M storage below-the-line allocation; above-the-line defaults.
REGIONX=(1M)	1 M storage below-the-line allocation; above-the-line defaults.
REGIONX=(,60M)	Storage below-the-line default; 60 M storage above-the-line allocation.
REGIONX=(5M,1G)	5 M storage below-the-line allocation; 1 G storage above-the-line allocation.

You can code up to five numeric digits and a unit specification for each allocation. K and M are allowed for below-the-line storage, and K, M, and G are allowed for above-the-line storage. In the case of REGIONX on an EXEC statement, either or both of these values can be defaulted to the REGIONX values on the JOB statement if it was specified.

Also, coding REGIONX in your EXEC JCL statement causes it to override any REGIONX values that are defined in the JOB statement. This configuration allows storage requirements to be tailored to a particular job step while allowing the JOB statement specification to act as a default when REGIONX= is not specified on an EXEC statement.

The Example 2-1 shows a sample JCL with REGIONX coded on JOB and EXEC statements. STEP2 requests 4 M of storage below-the-line, which overrides the JOB default of 2 M.

Example 2-1 REGIONX override

```
//JOBD JOB TIME=NOLIMIT,MSGCLASS=A,CLASS=A,NOTIFY=&SYSUID,
//         REGIONX=2M
//STEP1    EXEC  PGM=ADRDSSU
//SYSPRINT DD    SYSOUT=A
//DASD     DD    UNIT=3390,VOL=(PRIVATE,SER=BH5ST5),DISP=OLD
//TAPE     DD    UNIT=(VT3590),
//  LABEL=(1,SL),DISP=(NEW,CATLG),DSNAME=KWRES08.DUMP.BH5ST5
//SYSIN    DD    *
 DUMP  INDDNAME(DASD) OUTDDNAME(TAPE)
//STEP2    EXEC  PGM=ADRDSSU,REGIONX=4M
//SYSPRINT DD    SYSOUT=A
//DASD     DD    UNIT=3390,VOL=(PRIVATE,SER=BH5ST6),DISP=OLD
//TAPE     DD    UNIT=(VT3590),
//  LABEL=(1,SL),DISP=(NEW,CATLG),DSNAME=KWRES08.DUMP.BH5ST6
//SYSIN    DD    *
 DUMP  INDDNAME(DASD) OUTDDNAME(TAPE)
```

The REGIONX keyword can be used in JOBs and STEPs, including JOBs and procedures that are started by the **START** command. In these cases, the following scenarios are possible:

- For a started job, a REGION or REGIONX specification on the **START** command can replace a REGION or REGIONX specification on the JOB statement.
- For a started procedure, a REGION or REGIONX specification on the **START** command can replace a REGION specification on the JOB statement that is generated internally by **START** command processing.

2.2.1 New IEFUSI example

A new sample SMF Step Initiation Exit routine is available on SYS1.PARMLIB to assist customers in configuring their exits to handle REGION and REGIONX keyword.

Customers must rename it to IEFUSI, then customize, assemble, link, and place it into LPALIB so that it can be used.

3

Enhancements in opening data sets

This chapter describes z/OS V2R2 enhancements on reducing the allocation time.

3.1 Current open data set limits

Application and data growth often translates into a larger and increased number of DB2 tables to handle the amount of data that is created. In z/OS V2R1, up to 200,000 linear data sets can be kept open at a time by DB2. The time to open and close data sets tends to grow when large numbers of data sets are remained open. While there is no change in maximun numbers of open data set in DB2 with z/OS V2R2, z/OS V2R2 provides a reduction in time to open and close data sets when large numbers of data sets are accessed.

3.1.1 Changes in z/OS V2R2

Several changes were implemented to allocation routines to increase the maximum number of open data sets and enhance allocation times.

Now, all bits in SWA Virtual Addresses (SVAs) are used and redefined to be interpreted as an index and not as an offset. This change directly indexes to the appropriate SWA block address look-up table, which reduces elapsed time to open data sets.

In addition, SWA management infrastructure and the allocation's SYSDSN management blocks were moved to storage above the *bar*, which releases space below the bar for application usage.

SYSZTIOT was also updated to shorten exclusive hold time, which improves parallelism.

Usage and invocation

These changes are automatic and transparent to users. IEFQMREQ and SWAREQ services use the new infrastructure and any program can see improved allocation times, if the following conditions are met:

- It uses the S99DXACU (unauth) or S99TIOEX + S99DSABA + S99ACUCB (auth) SVC99 options.
- It provides its own DD names instead of system-generated.
- There are multiple tasks starting SVC99 in parallel.

Functions that are provided in older releases can also be used to improve allocation performance, including S99DASUP and MEMDSENQMGMT.

Migration and coexistence

In a mixed z/OS V2R1 / z/OS V2R2 sysplex environment, customers must ensure that OA45946 and OA46874 are installed on their z/OS V2R1 systems if they intend to use the JES2 INTERPRET=JES function.

4

Dynamic APF SMF Record

This chapter describes the new System Management Facilities (SMF) record that is used to track authorized program facility (APF) changes that are performed dynamically after an initial program load (IPL).

4.1 Dynamic APF SMF record enhancement

When APF commands are dynamically issued to change APF status for data sets, these changes are not tracked by SMF records. Therefore, changes cannot be monitored.

Starting with z/OS V2R2, a new SMF type 90, subtype 37 record is created whenever a dynamic APF change occurs. This solution provides improved auditability to APF changes.

The SMF record is created for each post-IPL APF change (APF statement in PROGxx, SETPROG APF command, and CSVAPF macro). IPL APF definitions are not tracked by SMF. The following event types are available:

- Add
- Delete
- Change to dynamic format
- Change to static format

The SMF record includes the following fields:

- Function
- Origin (CSVAPF, SET PROG, SETPROG with PROGxx suffix)
- Data set name
- Volume ID
- Timestamp of the operation
- Jobname of the requester
- CHKEY (for example, step name) of the requester
- Console ID of the requester
- Security product user token (Utoken) of the requester

5

Initial program load device number and volume

This chapter describes the new message that is used to identify the initial program load (IPL) device and volume.

5.1 IPL device number and volume message

Initially loading an incorrect volume is difficult to detect and can cause errors and slow down the IPL process.

A new message is provided in z/OS V2R2 that includes the wanted device number and volser information, which makes it easier to detect an error. Also, it is easier to associate an archived syslog with a specific set of maintenance on a volume.

Example 5-1 shows the new message that is displayed during IPL, where:

- `sdddd` is the subchannel set ID and device number that is used for the IPL
- `vvvvvv` is the volume name of the IPL device

Example 5-1 Sample IOS128I message

```
IOS128I IPL DEVICE: sdddd  VOLUME: vvvvvv
```

6

Infoprint enhancements

This chapter provides describes the enhancements that were made to Infoprint in z/OS V2R2 and includes the following topics:

6.1 TSO/E command for stopping a printer

Starting with z/OS V2R2, TSO users can issue TSO/E commands to start and stop Infoprint Server-managed output devices. These commands can also be used by an automation tool to perform such tasks.

The new **AOPCMND** command can be issued from the TSO/E READY prompt, ISPF command shell, batch job, or a started task to start and stop output devices. The command that is shown in Example 6-1 can be used to start a printer that is called “testprinter”.

Example 6-1 testprinter command

```
aopcmnd printer ‘testprinter’ start
```

Example 6-2 shows the complete syntax of the **AOPCMND** command (underscored values are the default).

Example 6-2 AOPCMND syntax

```
AOPCMND PRINTER 'printer_definition_name'
{START | STOP [(COMPLETE | HOLD | DELETE)] }
[INVENTORY(inventory_name)]
[XCFQUALIFIER(group_qualifier)]
```

You can use the TSO/E command **AOPCMND** to start and stop IP IBM PrintWay™ Extended mode printers instead of the use of Infoprint Central. To use the **AOPCMND** command, authorize the TSO/E command processor by editing the IKJTSOxx member of SYS1.PARMLIB and add **AOPCMND** to the AUTHCMD list.

6.2 Start and stop procedures for Infoprint Server daemons

Infoprint Server allows the use of JCL procedures to start and stop individual Infoprint Server daemons. System administrators can automate start and recovery actions by using z/OS automation options and simplify operation and procedures for managing Infoprint Server.

In z/OS V2R2, Infoprint Server is including new JCL procedures that can be customized to start and stop some or all of Infoprint Server daemons and obtain Infoprint Server status information.

The following JCL procedures are included in SYS1.IBM.PROCLIB:

- AOPDEMON

  Starts an individual Infoprint Server daemon; for example: `START AOPDEMON, TYPE=SSI`
- AOPSTAR2

  Starts one or multiple Infoprint Server daemons; for example: `START AOPSTAR2`
- AOPSTOP2

  Stops one or multiple Infoprint Server daemons; for example: `START AOPSTOP2`
- AOPSTAT

  Displays information about and the status of Infoprint Server; for example: `START AOPSTAT`

To use these new procedures, you must ensure that Dynamic configuration is enabled and the Operating Mode field is set to z/OS 2.2 in the ISPF System Configuration panel.

Operating Mode is a new element that was introduced with V2R2 with which users can decide whether new product functions are activated during or after installation.

When Dynamic Configuration is enabled, the element can be set by using the ISPF System Configuration panel (default value is set to z/OS V2R1). When Operating Mode is set to z/OS V2R2, the aopstart EXEC and startdaemons attribute in `aopd.conf` are ignored.

6.3 Variable mail inline message

Infoprint Server is providing a new function in IP Printway Extended mode with which personalized text can be included in each step of a batch job that is used to send print output via email.

By using this personalization, users can insert customized greeting (such as, “Dear Mr. Jones”) ahead of generated email or a standard email text that accompanies an attachment.

IP Printway Extended mode can include a text job attribute inline as a text string on the first line of an email. This function can be requested with the Inline text attribute field in the printer definition. The new keyword can be used in the following ways:

- For a batch job, with the JCL parameter or the `PRTATTRS` parameter in the JCL:

  ```
  NAME=’Dear Mr. Jones,’
  PRTATTRS=’name-text="Dear Mr. Jones,"’
  ```

- For an IP or AOPPRINT job, with a `-o` parameter:

  ```
  o ’name-text="Dear Mr. Jones,"’
  ```

- For a NetSpool job, in a data stream as a NetSpool Job Attribute:

  ```
  <<IBMJOBATTR0020name-text=Dear Mr. Jones,
  ```

Example 6-3 shows a JCL example that instructs IP Printway Extended mode to include the name or department in the first line of the email.

Example 6-3 Sample JCL

```
//NAMEOUT OUTPUT DEST=xxx,
// NAME='Dear Mr. Jones',
// PRTATTRS=’mail-inline-text-attribute="name-text"'
//....
```

6.4 Search output from TSO jobs by using Infoprint Central

Infoprint Server is providing a new function in Infoprint Central to allow searching for output from TSO jobs among JES print jobs.

You can now search for these jobs by using the Work with Print Jobs tab in Infoprint Central and then selecting **Find** to search any job, including output from TSO jobs. This change applies when searching for JES Print Jobs only.

In JES2, TSO output jobs often have a job ID prefix of TSU.

6.5 Infoprint Central redesign and IBM HTTP Server

Starting with this release, support was removed for IBM HTTP Server Powered by Domino®. Infoprint Central requires changes to work with IBM HTTP Server – Powered by Apache (31-bit version).

To use Infoprint Central, the configuration of IBM HTTP Server – Powered by Apache must be customized, which is shipped with the release.

The IBM HTTP Server – Powered by Apache files are shipped with z/OS V2R2. The default location is: `/usr/lpp/ihsa_zos/.31bit/`.

Edit and update configuration files for the IBM HTTP Server – Powered by Apache. These files are in the target directory where the HTTP Server was installed. You must make this change before you start the HTTP Server. The following configuration file contains directives that customize the HTTP Server. Add directives to the configuration file so that the HTTP Server can display Infoprint Central web pages, as shown in the following example:

`conf/httpd.conf`

When IBM HTTP Server – Powered by Apache is installed, the default environment variable file `bin/envvars` is created. To use the IBM HTTP Server – Powered by Apache with Infoprint Central, you must customize the environment variable file by adding Infoprint Server-related environment variables at the end of the file.

For more information about Infoprint configuration, see *z/OS Infoprint Server Customization Version 2 Release 2*, SA38-0691.

7

GRS EQDQ Monitor

This chapter describes the new enhancements to GRS EQDQ that monitor capture ENQ/DEQ diagnostics for problem determination while minimizing the effect to system performance.

7.1 EQDQ Monitor overview

Monitoring and diagnosing ENQ-related errors in z/OS might be difficult without built-in tracing options. The use of GRS EQDQ monitor (ISGAUDIT) provides tracing at the cost of ENQ performance.

New enhancements to GRS EQDQ monitor capture ENQ/DEQ diagnostics for problem determination while minimizing the effect to system performance. It also uses SMF 87 and is more cohesive, understandable filtering than EQDQ Monitor.

7.1.1 Changes in z/OS V2R2

Before you can use new GRS EQDQ capabilities, you must activate SMF Tracing of Type 87 in SMFPRMxx. The following subtypes that can be traced for specific information:

- SYS(TYPE(87(1))) for subtype 1 only: QSCAN
- SYS(TYPE(87(2))) for subtype 2 only: ENQ/DEQ
- SYS(TYPE(87)) for all subtypes

SMF 87 Recording in V2R2 drives SMF exit IEFU84 switching from IEFU85 with OA42221.

In z/OS V2R2, SMF 87 records are no longer cut under the caller's thread. Instead, SMF type 87 subtype 1 is cut whenever global generic queue scan is issued while subtype 2 has multiple ENQ/DEQ requests per record that are cut within one SMF interval.

Because new records are tracked by SMF, consider planning SMF resources for the new data stream and consider the use of a separate log stream for type 87s to avoid running out of space for other types because SMF 87 subtype 2s can be cut in much greater frequency.

The following subtypes are available in SMF 87:

- SMF 87 Subtype 1 - QSCAN:
  - SMF87RHS: Header
  - SMF87DEF: Self-defining section
  - SMF87REQ: Requester section
  - SMF87QSCAN: Queue scan section

  These subtypes are cut whenever global generic queue scan is issued and are introduced to help track queue scan abusers.
- SMF 87 Subtype 2 - ENQ/DEQ:
  - SMF87RHS: Header section
  - SMF87DEF: Self-defining section
  - SMF87REQ 1 - n: Requester section; includes items, such as PSW, JOBNAME, ASID, TCB, and time of request. New fields added that are specific to ENQ/DEQ.
  - SMF87ENQ 1 - n: Data Section; includes QNAME, RNAME, SCOPE, and other information about the request parameters.

You can set up the filters you want to include or exclude from monitoring in GRSMONxx member. The * and ? wildcard characters can be used in the definition of the filters and the filters can be defined in any sequence. Also, Exclude list trumps Include list.

Example 7-1 shows a sample configuration of GRSMONxx member. Long RNAMEs are supported up to 255 characters.

Example 7-1 Sample GRSMONxx member

```
FILTER INCLUDE ENQDEQ QNAME(SYSDSN) RNAME(*)
      SCOPE(SYSTEMS)
FILTER EXCLUDE ENQDEQ QNAME(*) RNAME(*)
      JOBNAME(MYJOB)
FILTER INCLUDE QSCAN GENERIC(YES) XSYS(YES)
FILTER INCLUDE ENQ QNAME(*) RNAME(*)
      RNLMATCH(YES)
```

Note: Avoid FILTER INCLUDE ENQ QNAME(*) RNAME(*) without other modifier because it produces too much data. For more information, see SYS1.SAMPLIB(ISGMON00).

After you enable SMF type 87 records and define the monitor filters, you can control GRS monitoring by using one of the following commands:

- SETGRS MONITOR=YES

  Enables the GRS Monitoring with existing filtering.
- SETGRS MONITOR=NO

  Disables the GRS monitoring.
- SETGRS GRSMON=xx

  Changes the active filters and activates, where xx corresponds to GRSMONxx.
- MONITOR(YES|NO) keyword in GRSCNFxx

  No GRSMON(xx) analog in GRSCNFxx.
- Display GRS[,SYSTEM]

  Shows MONITOR status and current GRSMONxx.

There are no migration and coexistence considerations because this solution is a single-system solution.

8

BCPii System Management Facilities recording

This chapter describes System Management Facilities (SMF) recording capability for BCPii changes to hardware resources.

8.1 BCPii overview

The Base Control Program internal interface (BCPii) allows authorized z/OS applications to have Hardware Management Console (HMC)-like control over systems in the process control (HMC) network. It provides a complete communication isolation of networks (intranet and internet) from the process control (HMC) network.

You can use HWISET API to set Support Element (SE) and HMC attributes. You can also use HWICMD API to perform commands against an HMC-managed object that is associated with CPCs and CPC images (LPARs).

8.2 New SMF type 106 record

New to z/OS V2R2, SMF type 106 records are now written when a HWISET or HWICMD APIs are successfully called. This function provides sufficient audit information to know what resources were modified by BCPii applications.

The SMF type 106 record holds information about Connection Type of the HWISET or HWICMD request, CPC Name, Request parameter, ASID, Job name, and User ID. The following subtype records (each to cover one API) are available:

- Subtype 1 provides information about HWISET, including Set type (resource that was modified), Set Type Value Length (length of the value being set), and Set Parameter (the actual value that is set).
- Subtype 2 provides information about HWICMD, including Command Type (command that was issued), Command parameter list passed to BCPii, and Optional XML data sent on request.

To activate SMF recording, you must add the necessary statements to the SMFPRMxx parmlib member for SMF Type 106, and issue the **SET SMF=xx** command to activate the parmlib changes. Example 8-1 shows some possible configurations for SMFPRMxx member.

Example 8-1 Sample SMFPRMxx configurations

```
SYS(TYPE(106))
SYS(TYPE(106(1)))
SYS(TYPE(106(2)))
SYS(TYPE(106(1:2)))
```

A mapping macro for SMF type 106 is available at SYS1.MACLIB, member HWISMF6A. The following sample formatting JCL is available to handle SMF records, which are available on SYS1.SAMPLIB:

- HWI6AFMT: Copies BCPii SMF Type 106 records from a data set or logstream to a temporary dataset. By using the DFSORT-provided ICETOOL, it sorts the type 106, subtype 1, and subtype 2 records and produces a summary report for the type 106 records and a detailed report for type 106 for subtype 1 and subtype 2.
- HWIRPTMP: SMF type 106 JCL variable map as input to the DFSOR-provided ICETOOL.

9

Provisioning based on CPU consumption

This chapter describes provisioning that based on CPU consumption, which provides the means to promptly react to high CPC-wide CPU consumption.

9.1 Capacity Provisioning overview

Customers can define Capacity Provisioning Policies for provisioning On Off Capacity on Demand (OOCoD) processor capacity through scheduled activations for anticipated capacity bottlenecks. The policies also can be defined for workload-based activations that are driven by WLM service classes that exceed certain PI limits for unexpected capacity shortages.

In situations where customers consider the central processor complex (CPC)-wide CPU consumption as the primary and sufficient evidence for a processor capacity shortage and cannot identify suitable WLM service classes that can trigger provisioning, the new provisioning-based on CPU consumption provides the means to promptly react to high CPC-wide CPU consumption.

9.1.1 Changes in z/OS V2R2

Specifications for how Provisioning Manager must react to high CPC-wide CPU consumption require the following definitions in Capacity Provisioning:

- Domain Configuration

  Defines a Capacity Provisioning management domain that specifies CPCs with processor capacity to be managed and systems in those CPCs that are used to observe the CPU consumption.

- Policy

  Defines scopes and rules for changes of Processor Capacity, where the following conditions apply:

  - Processor Scopes define the total number of allowed changes
  - Time Conditions define when increases and decreases of capacity can occur
  - Utilization Conditions specify when a CPC is constraint and should start a step-by-step increase and decrease of processor capacity

The new Policy supporting Utilization Based Provisioning can be defined only in z/OSMF for z/OS V2R2.

Installation and usage

This section provides a step-by-step basic configuration of Capacity Provisioning.

Domain Configuration

To use Capacity Provisioning-based on CPU usage, managed CPCs must be specified, including a valid OOCoD Record ID. Also, at least one system on the managed CPC must be defined to observe the CPC-wide CPU consumption metrics.

Figure 9-1 shows a sample CPC and Systems configuration from z/OSMF.

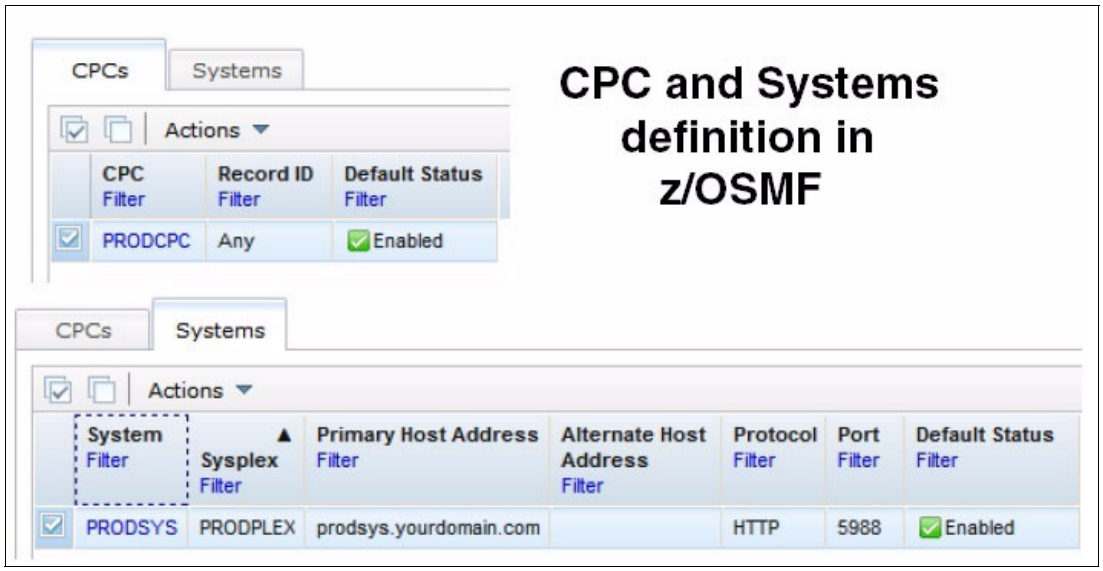


Figure 9-1 CPC and systems definition

Policy Definitions

Before defining your policy, you must consider and answer the following questions regarding CPU and capacity management:

- How much more Processor Capacity should be managed?
- Where and what kind of capacity should be managed?
- When and in which situations should be provisioned?

The information about how much more Processor Capacity is needed is used to define the maximum extra capacity the Provisioning Manager is allowed to manage and to set up step-by-step increments for utilization-triggered or workload-triggered management.

Use Maximum Processor Scope and Rule's Processor Scope options to define how much more processor capacity can be provisioned at most. These settings apply to all defined scheduled activations, workload-based activations, and utilization-based activations. Combinations of scheduled activations, workload-based activations, and utilization-based activations never exceed the maximum activation level.

Figure 9-2 shows a sample configuration for Maximum Processor Scope tab.

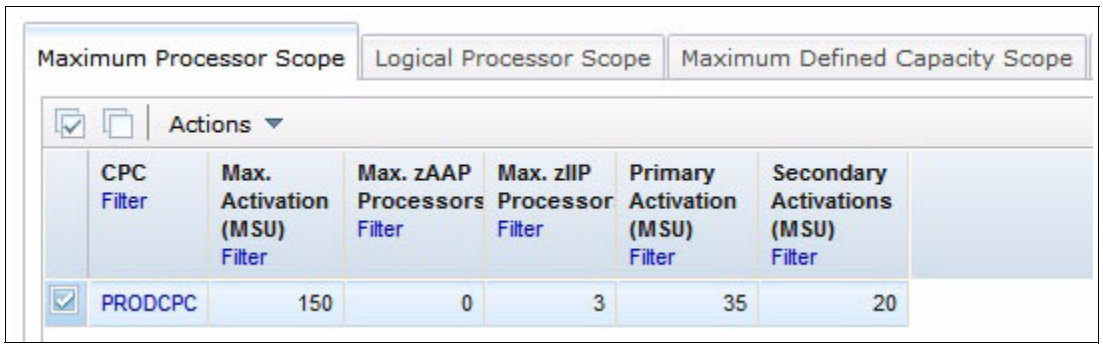


Figure 9-2 Maximum Processor Scope tab

Then, define where and what kind of capacity must be managed in which situation by using the Utilization Condition tab. This tab can be defined with Workload Condition.

Figure 9-3 shows a sample Utilization Conditions definition.

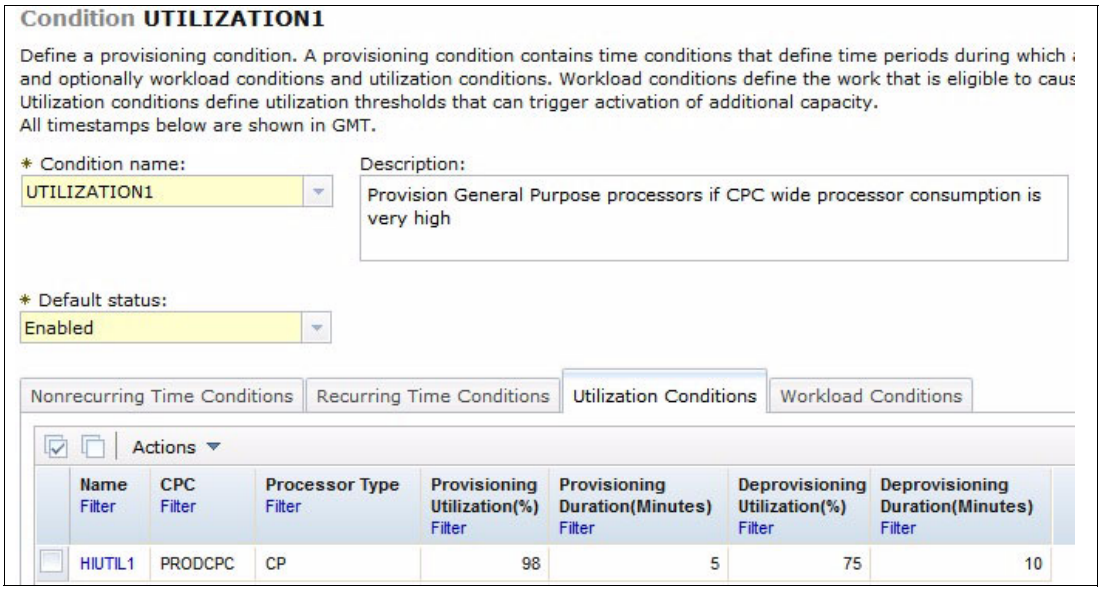


Figure 9-3 Sample Utilization Conditions definition

Next, define where capacity must be managed in one of the following components:

- A CPC that is listed in the active Domain Configuration and with a defined Processor Limit
- Any CPC to apply this condition to all CPCs of the active Domain Configuration and with a defined Processor Limit

The Provisioning Manager automatically observes suitable systems that are running on that CPC to monitor CPC-wide CPU consumption data.

Also, define the following type of processor that must be managed:

- General Purpose (CP)
- zIIP
- zAAP

Finally, define the following situations that qualify for provisioning and deprovisioning extra processor capacity, as shown in Figure 9-4 on page 29:

- Provisioning utilization (%)

  The CPC-wide CPU consumption level of the specific processor type that must be exceeded to trigger provisioning.

- Provisioning duration (minutes)

  The minimum time during which the CPC-wide CPU consumption must exceed the provisioning utilization to trigger provisioning.

- Deprovisioning utilization (%)

  The CPC-wide CPU consumption level of the specific processor type must fall below this limit to trigger deprovisioning of more processor capacity.

- Deprovisioning duration (minutes)

  The minimum time during which the CPC-wide CPU consumption must fall below the deprovisioning utilization to trigger deprovisioning.

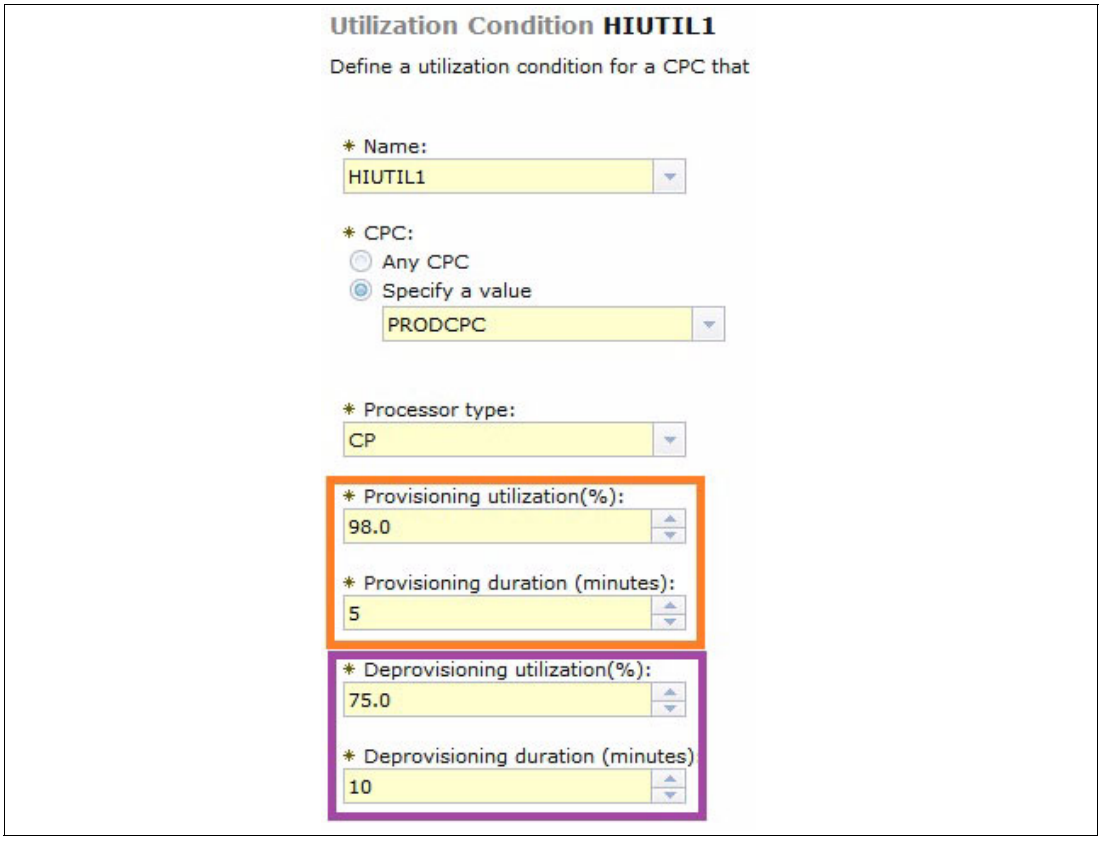


Figure 9-4 Define utilization options

Unlike the workload condition, the utilization condition-driven activations are triggered by any workload that increases up the CPC-wide CPU consumption.

Provisioning Manager messages on the console about processor capacity changes it starts and of their successful implementation. It does not inform about the starting action or policy element. Example 9-1 shows a sample message.

Example 9-1 Sample CPO4108I message

```
CPO4108I Activation of resources for CPC ECL2 successfully initiated:
model 729 (3/0) with 2 zAAPs and 2 zIIPs
CPO3030I Command completed successfully for CPC ECL2
```

Reporting

The following reports can be created to monitor Capacity Provisioning activation:

- Monitored CPCs and their relevant processor capacity data
- Current policy with capacity scopes, time frames, and condition settings
- Situations in which utilization qualifies for step-by-step changes of processor capacity

- Current processor capacity and Provisioning Manager status regarding a specific CPC and OOCoD Record ID
- Listing of all capacity changes that are started by the Provisioning Manager

The following console commands provide more information about CPU consumption, processor capacity, and policy. This information also can be gathered from z/OSMF:

- MODIFY CPOSERV,APPL=REPORT CONFIGURATION

  Displays the current Processor Capacity of a CPC.
- MODIFY CPOSERV,APPL=REPORT POLICY

  Displays the active policy's utilization conditions.
- MODIFY CPOSERV,APPL=REPORT UTILIZATION CPC=ECL2

  Displays the CPU consumption and detected demands for increasing processor capacity.
- REPORT UTILIZATION CPC=*

  Displays utilization report for all observed CPCs.

10

System REXX enhancements

System REXX (also known as SYSREXX) is a z/OS component that allows REXX execs to be run outside of conventional TSO/E and batch environments.

System REXX execs can be started by using an operator command; for example, it can start IBM MVS™ system commands and receive results in various ways.

This chapter describes new support and enhancements for SYSREXX in z/OS V2R2 and includes the following topics:

10.1 New functions and interfaces for SYSREXX

SYSREXX provides the following enhancements in z/OS V2R2:

- New operator commands:
  - Cancel command for SYSREXX execs
  - End AXRnn address spaces
  - Run operator-started TSO=NO SYSREXX execs
  - A clean STOP AXR command support
- The following new parmlib customization keywords are available:
  - A **TimeInt** keyword to override the time interval default of 30 seconds
  - A **MaxTsoServers** keyword to change the default of eight servers to a value of 1 - 16.
- A new **AXRWTOR** function provides the ability for an exec to issue a write to operator with reply (WTOR) and receive the response.

10.1.1 New operator cancel command

You can start a SYREXX exec by using the new OREQTOKEN option, as shown in Figure 10-1.

```
MODIFY AXR,rexxname,OREQ
```

Figure 10-1 Start a SYREXX exec by using the new OREQTOKEN option

Note: A request token uniquely identifies an in-progress exec.

You can use the request token that is displayed in this case to cancel the REXX exec, as shown in Figure 10-2.

```
MODIFY AXR,SR CANCEL,REQTOKEN=reqtoken
```

Figure 10-2 Cancel a SYSREXX exec by using its request token

10.1.2 Stopping all TSO Server address spaces

You can use the **MODIFY** command that is shown in Figure 10-3 to stop all TSO Server address spaces.

```
MODIFY AXR,SR STOPTSO
```

Figure 10-3 Stopping all TSO Server address spaces

Consider the following points:

- Running this command still allows running execs to complete. It also ends the execs if they do not complete within a time threshold (approximately 30 seconds to 1 minute).
- Subsequent AXREXX TSO=YES requests fail with return code 0C and reason code xxxx0C11.

10.1.3 Starting TSO Server address spaces

A new **STARTTSO** command is available that is the inverse of **STOPTSO**, as shown in Figure 10-4.

```
MODIFY AXR,SR STARTTSO
```

Figure 10-4 Starting TSO Server address spaces

Note: This command also can be used to interrupt an in-progress STOPTSO processing.

10.1.4 New TSO parameter on running a REXX

Figure 10-5 shows how to use the new TSO parameter for running a SYSREXX exec with or without TSO setup.

```
MODIFY AXR,rexxname,TSO=<NO|YES>
```

Figure 10-5 Use of new TSO parameter on running a SYSREXX exec

The default setting for TSO is YES.

Note: Operator started TSO=NO execution might be used to run a REXX procedure without POSIX, console, and TSO commands.

10.1.5 Stopping System REXX

In z/OS V2R2, SYSREXX can be stopped by using the **STOP** command, as shown in Figure 10-6.

```
STOP AXR
```

Figure 10-6 Stopping system REXX

Consider the following points:

- This command provides a graceful end that is a much better alternative than the use of the **FORCE AXR,ARM** command.
- It waits for all in progress execs to complete and ends the execs if they do not complete within a time threshold (of about 30 seconds to 1 minute).
- New SYSREXX execs no longer can be started.
- An operator can still issue **MODIFY AXR** commands while STOP processing is in progress.

10.1.6 New and changed parmlib options

In z/OS V2R2, the following new parmlib keywords were introduced:

- Option TIMEINT sets the default timeout interval that a SYSREXX exec can run until it is ended.

  It can be used to override the default of 30 seconds.

You can specify a value of 0 - 21474536. A value of 0 means that SYSREXX execs do not time out by default.

- Option MAXTSOSERVERS sets the maximum of TSO Servers.

  You can specify a value of 1 - 16. This value overrides the current default of 8.

- Option MAXWORKERTASKS limit was changed.

  Previously, 4 - 64 tasks were supported. Now, 4 - 32 tasks are supported.

  The default is still 32. An error message is issued if the value specified exceeds 32 and the default is used.

10.1.7 New AXRWTOR function

The new function AXRWTOR provides the ability for an exec to issue a WTOR and receive the response.

Consider the following points:

- The message routing is identical to that of function AXRWTO.
- The message is sent to console name as specified in the CONSDATA keyword of the AXREXX invocation. If CONSDATA is not specified, system default routing attributes apply.

10.1.8 Dependency on RACF passphrase support

Internal EMCS consoles are started at System REXX initialization time instead of when a worker task is created for TSO=NO.

Important: Potential deadlocks on SYSZMCS must be avoided when AXREXX is started during CONSOLE LOGON. A possible deadlock still exists for TSO=YES. Therefore, you should not use TSO=YES for parsing IBM RACF® passphrases.

10.2 Use of the new functions and interfaces

In this section, several examples of the new functions are described.

10.2.1 Canceling a SYSREXX exec by using a request token

Figure 10-7 on page 35 shows the use of the new OREQTOKEN option and the use of the token to cancel a SYSREXX exec.

```
F AXR,SLEEPING,T=300,OREQ
AXR0213I EXEC NAME=SLEEPING 768
REQTOKEN=00004000000000000CF5D33CE33745CC5
sleeping...
F AXR,SR CANCEL,REQTOKEN=00004000000000000CF5D33CE33745CC5
AXR0208I SYSREXX CANCEL OF REQTOKEN=00004000000000000CF5D33CE33745CC5
COMPLETED SUCCESSFULLY.  EXEC NAME IS SLEEPING
BPXP018I THREAD 13E2480000000001, IN PROCESS 33620842, ENDED 773
WITHOUT BEING UNDUBBED WITH COMPLETION CODE 0013E000
, AND REASON CODE 00000000.
AXR0203I AXREXX INVOCATION OF SLEEPING FAILED. 774
RETCODE=0000000C RSNCODE=05050C07
REQTOKEN=00004000000000000CF5D33CE33745CC5
DIAG1=00000000 DIAG2=00000000 DIAG3=00000000 DIAG4=00000000
```

Figure 10-7 Use of the OREQTOKEN option and cancel a SYSREXX by using the token

Note: Return code C and reason code 0C07 means that the request was canceled.

10.2.2 Stopping and restarting TSO Server an address spaces

Figure 10-8 shows excerpts of the SYSREXX status, including the current TSO Server address spaces.

```
F AXR,SR ST
AXR0200I SYSREXX STATUS DISPLAY 776
 SYSTEM REXX STARTED AT 09.32.58 ON 07/29/2015
 PARMLIB MEMBERS:      AXR00
 CPF:  § (SYSPLEX)      AXRUSER:  IBMUSER
 TIMEINT:       30     TMP: NOT ENABLED
 SUBSYSTEM:   AXR      TSO=YES ENABLED
...
 TSO SERVER SPACES:    ACTIVE:   0     TOTAL:   1
                       IDLE:     1     MAX:     8
...
```

Figure 10-8 Excerpts of the SYSREXX status

Figure 10-9 shows stopping the TSO Server address spaces.

```
F AXR,SR STOPTSO
AXR0214I SYSREXX STOPTSO IS ACCEPTED.  ALL SUBSEQUENT TSO=YES
REQUESTS WILL BE REJECTED
...
AXR0209I SYSREXX STOPTSO COMMAND COMPLETE.  ISSUE SYSREXX STARTTSO TO
RESUME AXREXX TSO=YES PROCESSING
```

Figure 10-9 Stopping TSO Server address spaces

As shown in Figure 10-10, the new status excerpts show that TSO=YES is disabled.

```
 F AXR,SR ST
 AXR0200I SYSREXX STATUS DISPLAY 786
  SYSTEM REXX STARTED AT 09.32.58 ON 07/29/2015
  PARMLIB MEMBERS:      AXR00
  CPF:  § (SYSPLEX)      AXRUSER:  IBMUSER
  TIMEINT:       30     TMP: NOT ENABLED
  SUBSYSTEM:   AXR      TSO=YES DISABLED
...
  TSO SERVER SPACES:    ACTIVE:   0     TOTAL:   0
                        IDLE:     0     MAX:     8
...
```

Figure 10-10 SYSREXX status shows TSO=YES being disabled

How to restart TSO Servers and receive TSO=YES is shown Figure 10-11.

```
 F AXR,SR STARTTSO
 AXR0211I AXREXX TSO=YES PROCESSING IS RESUMED
 F AXR,SR ST
 AXR0200I SYSREXX STATUS DISPLAY 791
  SYSTEM REXX STARTED AT 09.32.58 ON 07/29/2015
  PARMLIB MEMBERS:      AXR00
  CPF:  § (SYSPLEX)      AXRUSER:  IBMUSER
  TIMEINT:       30     TMP: NOT ENABLED
  SUBSYSTEM:   AXR      TSO=YES ENABLED
...
  TSO SERVER SPACES:    ACTIVE:   0     TOTAL:   0
                        IDLE:     0     MAX:     8
...
```

Figure 10-11 Restarting TSO Server address spaces

10.2.3 Stopping and restarting SYSREXX

In z/OS V2R1, the new SYSREXX function was available to run under the Terminal Monitor Program (TMP) to retrieve unsolicited syslog messages, if wanted. To use the TMP instead of the TSO Environment Service, **AXRRXWKD** must be set up as an authorized command in IKJTSOxx.

Attention: This set up must occur before System REXX initialization.

As shown in Figure 10-11, TMP is listed as not enabled. It is assumed **AXRRXWKD** was updated and activated as authorized and needed.

Note: The new **stop** command cleanly recycles SYSREXX and enables the use of TMP.

Stopping and restarting SYSREXX is shown in Figure 10-12.

```
 P AXR
 AXR0218I STOP AXR COMMAND COMPLETE
 AXR0116I SYSTEM REXX IS TERMINATING
...
 IEA631I  OPERATOR *AXR0174 NOW INACTIVE, SYSTEM=SC74    , LU=AXREMCS
...
 IEF196I IEF142I IEESYSAS AXR - STEP WAS EXECUTED - COND CODE 0000
 F AXR,SR ST
 IEE341I AXR               NOT ACTIVE
 S AXRPSTRT
...
 $HASP373 AXRPSTRT STARTED
 $HASP395 AXRPSTRT ENDED - RC=0000
...
 IEE252I MEMBER  AXR00   FOUND IN SYS1.PARMLIB
...
 IEA630I  OPERATOR *AXR0174 NOW ACTIVE,   SYSTEM=SC74    , LU=AXREMCS
...
 AXR0102I SYSTEM REXX INITIALIZATION COMPLETE
```

Figure 10-12 Stopping and restarting SYSREXX

Figure 10-13 shows TMP as enabled.

```
 F AXR,SR ST
 AXR0200I SYSREXX STATUS DISPLAY 969
  SYSTEM REXX STARTED AT 22.34.04 ON 08/07/2015
  PARMLIB MEMBERS:      AXR00
  CPF:  § (SYSPLEX)      AXRUSER:  IBMUSER
  TIMEINT:       30     TMP: ENABLED
  SUBSYSTEM:   AXR      TSO=YES ENABLED
...
```

Figure 10-13 TMP shown as enabled via SYSREXX status display

10.2.4 Showing how to use the AXRWTOR function

Figure 10-14 shows how to issue a WTOR and receive the response.

```
 F AXR,SENDWTOR,T=0
/* REXX using WTOR */ 503
 connect_id = "FIRSTLINE"; lines = Sourceline()
 Do i=1 To lines; Call Axrmlwto Sourceline(i), "connect_id", "D"; End
 Call AXRMlwto, "connect_id", "E"
 Call AXRWtor "Reply to this message with any text."
 Call AXRWto "Replied data is:" AxrReply /* Reply stored in AxrReply */
 Exit
*017 Reply to this message with any text.
 R 17,'This is an arbitrary text.'
 IEE600I REPLY TO 017 IS;'This is an arbitrary text.'
 Replied data is: This is an arbitrary text.
```

Figure 10-14 Sending a WTOR and receiving the response

Related publications

The publications that are listed in this section are considered particularly suitable for a more detailed discussion of the topics that are covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide more information about the z/OS V2R2 updates. Some of the publications that are referenced in this list might be available in softcopy only:

- *z/OS V2R2: JES2, JES3, and SDSF,* SG24-8287
- *z/OS V2R2: Security,* SG24-8288
- *z/OS V2R2: Storage Management and Utilities,* SG24-8289
- *z/OS V2R2: Availability Management,* SG24-8290
- *z/OS V2R2: Performance,* SG24-8292
- *z/OS V2R2: Operations,* SG24-8305
- *z/OS V2R2: Diagnostics,* SG24-8306
- *z/OS V2R2: Sysplex,* SG24-8307
- *z/OS V2R2: UNIX System Services* SG24-8310
- *z/OS V2R2: User Interfaces,* SG24-8311
- *z/OS V2R2: IBM ServerPac*, SG24-8500

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

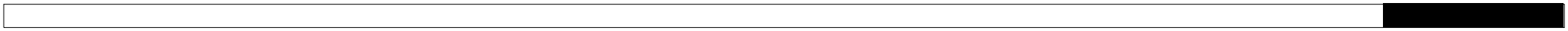
Help from IBM

IBM Support and downloads:

ibm.com/support

IBM Global Services:

ibm.com/services



(0.1”spine)
0.1”<->0.169”
53<->89 pages

SG24-8305-00

ISBN 0738441333

Printed in U.S.A.