Tips, Tricks, and Insights into Diagnosing Common IBM z/OS Errors

z/OS Systems Programming

February 25th, 2026

John C. Shebey III jshebey@us.ibm.com
IBM, STSM - z/OS Worldwide Technical Support

Agenda

Objectives

- Get started diagnosing common z/OS problems
- Search for matching APAR/problems
- Route to correct product/component
- Achieve greater understanding when working with z/OS Support

The purpose of this presentation is to help you get started diagnosing common z/OS issues, including ABENDs, messages, loops and hangs. You will learn about the tools necessary to identify symptoms for common problems, which can be used to search for matching APARs and to help route problems to the correct product/component.

OVERVIEW OF IPCS

What is IPCS?

- **IPCS** (**I**nteractive **P**roblem **C**ontrol **S**ystem)
  - Interactive tool provided with z/OS to aid in diagnosing software failures
  - Provides formatting and analysis support for dumps and traces produced by z/OS, other program products, and applications that run on z/OS
  - Can be invoked from ISPF or batch

IPCS can be run from the FOREGROUND (TSO Line Mode or ISPF Dialog Mode) or the BACKGROUND (BATCH Mode).

Data Viewable in IPCS

- Unformatted Dumps
  - SYSMDUMP
  - **SVC Dump**
  - SADUMP (Standalone Dump)
- Active Storage
  - Limited storage on active system (Source=ACTIVE)
- GTF/CTRACE Data Written to External Writer
  - IPCS Option 2.7 (ANALYSIS -> TRACES)
- DAE (Dump Analysis Elimination) Data
  - IPCS Option 3.5 (UTILITY -> DAE)

IPCS cannot be used to view formatted user dumps, such as SYSUDUMP and SYSABEND dumps. These types of dumps can be browsed in ISPF 3.4. IPCS is used to view unformatted user dumps (e.g. SYSMDUMP) and system dumps, including SVC Dumps and SADUMPs. IPCS Option 2.7 can be used to view trace data, and IPCS Option 3.5 can be used to view records in a DAE data set. It's even possible to view storage on the active system where IPCS is running!

IPCS Basic Navigation

- Main menu when IPCS dialog invoked from ISPF
- Customizable like other ISPF dialogs
  - Panels shipped by default in **SYS1.SBLSPNL0** pointed to by ISPPLIB

```
------------------------ IPCS PRIMARY OPTION MENU  -------
OPTION  ===>

    0  DEFAULTS     - Specify default dump and options
    1  BROWSE       - Browse dump data set
    2  ANALYSIS     - Analyze dump contents
    3  UTILITY      - Perform utility functions
    4  INVENTORY    - Inventory of problem data
    5  SUBMIT       - Submit problem analysis job to batch
    6  COMMAND      - Enter subcommand, CLIST or REXX exec
    T  TUTORIAL     - Learn how to use the IPCS dialog
    X  EXIT         - Terminate using log and list defaults
```


This is the main menu that's presented when IPCS is invoked in ISPF Dialog Mode. We'll discuss some of the options during this presentation.

Enter the OPTION # on the command line. You can jump to an OPTION from any IPCS command line by entering '=n', where 'n' is the OPTION #. For example, entering '=0' on the IPCS command line will jump to IPCS OPTION 0 - DEFAULTS.

IPCS Option 0: DEFAULTS

- Source: Specify dump name
- Scope: Set to BOTH and press ENTER

```
Command  ===>
  Scope    ==> BOTH     (LOCAL, GLOBAL, or BOTH)

  If you change the Source default, IPCS will display the current default
  Address Space for the new source and will ignore any data entered in
  the Address Space field.

  Source  ==> DSNAME('SHARE.S24601.DUMP1')
  Address Space   ==> ASID(X'0025')
  Message Routing ==> NOPRINT TERMINAL NOPDS
  Message Control ==> CONFIRM VERIFY FLAG(WARNING)
  Display Content ==> NOMACHINE REMARK REQUEST NOSTORAGE SYMBOL NOALIGN
```


LOCAL defaults: These values are currently in use for an ISPF screen in the IPCS dialog, for a batch IPCS session, or for an IPCS interactive line-mode session.

GLOBAL defaults: These values are used to establish the LOCAL defaults when IPCS processing starts in an ISPF screen, a batch IPCS session, or an IPCS interactive line-mode session.

Typically, specify Scope of BOTH. Once you press ENTER, the text “BOTH updated” appears in the upper right corner of the screen, but LOCAL reappears in the Scope setting.

IPCS Option 6: COMMAND

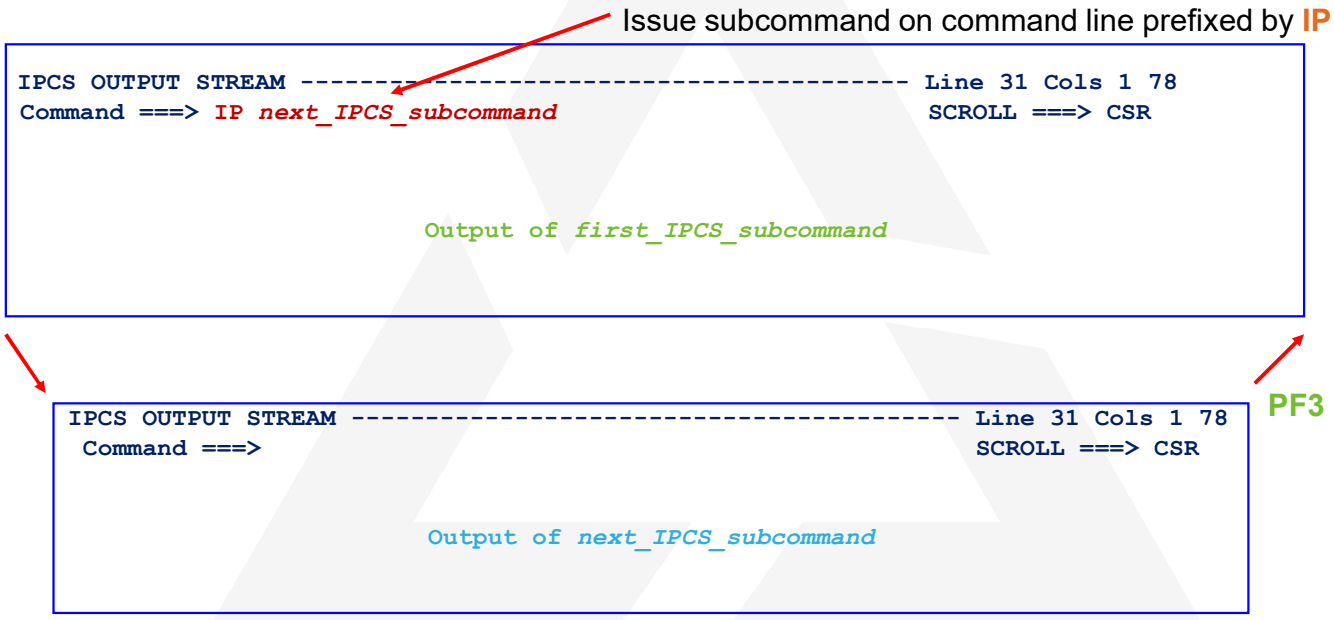
```
  Enter a free-form IPCS subcommand or a CLIST or REXX exec invocation below:

===> IPCS_subcommand

----------------------- IPCS Subcommands and Abbreviations --------------------
ADDDUMP             | DROPDUMP, DROPD    | LISTDUMP, LDMP     | RENUM,     REN
ANALYZE             | DROPMAP,  DROPM    | LISTMAP,  LMAP     | RUNCHAIN, RUNC
ARCHECK             | DROPSYM,  DROPS    | LISTSYM,  LSYM     | SCAN
ASCBEXIT, ASCBX     | EPTRACE            | LISTUCB,  LISTU    | SELECT
ASMCHECK, ASMK      | EQUATE,   EQU, EQ  | LITERAL            | SETDEF,    SETD
CBFORMAT, CBF       | FIND,     F        | LPAMAP             | STACK
CBSTAT              | FINDMOD,  FMOD     | MERGE              | STATUS,    ST
```


IPCS Subcommands can be entered from IPCS Option 6 (IPCS Subcommand Entry), or they can be issued on any command line with the prefix of IP. Details about these IPCS subcommands can be found in the z/OS MVS IPCS Commands manual.

Stacking IPCS Subcommands



To be able to stack IPCS subcommands, SYS1.SBLSTBL0 must be concatenated to ISPTLIB, and ISPF keyword NEWAPPL(BLSG) must be used when invoking IPCS initially.

The PF3 key can be used to un-stack the subcommand output and return you to your previous output screen.

Dump Initialization

- Issuing your first IPCS subcommand will cause the dump to initialize

```
IKJ56650I TIME-10:24:35 AM. CPU-00:00:00 SERVICE-13137 SESSION-02:50:11 MARCH 1,2024
BLS18122I Initialization in progress for DSNAME('SHARE.S24601.DUMP1')
BLS18124I TITLE=SHR2ESTA DETECTED ABEND S00C4, REASON 00000011
BLS18223I Dump written by z/OS 03.01.00-0 SVC dump - level same as IPCS level
```

- Initialization is not automatic from IPCS Option 0 (DEFAULTS)

Note that some IPCS subcommands do NOT result in the initialization of a dump. One example is the LIST TITLE command.

Note that the dump title is displayed (message BLS18124I) during dump initialization. Also, a message is issued to confirm that the z/OS release level of the dump matches the IPCS release level. The release levels must match to ensure that reports format correctly.

Dump Initialization: Summary Dump Data

- While dump is initializing, message BLS18160D may be displayed

```
 IKJ56650I TIME-10:24:35 AM. CPU-00:00:00 SERVICE-13137 SESSION-02:50:11 MARCH 1,2024
BLS18122I Initialization in progress for DSNAME('SHARE.S24601.DUMP1')
BLS18124I TITLE=SHR2ESTA DETECTED ABEND S00C4, REASON 00000011
BLS18223I Dump written by z/OS 03.01.00-0 SVC dump - level same as IPCS level
BLS18222I z/Architecture mode system
BLS18160D May summary dump data be used by dump access?  Enter Y to use, N to bypass.
Y
```

- Recommend replying '**Y**' to message BLS18160D

- Summary dump data includes:
  - Storage dumped closest to time of error
  - Storage pointed to by PSW/registers at time of error

You will only get message BLS18160D if 'CONFIRM' is specified in the defaults (IPCS Option 0). If 'NOCONFIRM' is specified, this message will not be issued, and summary dump data will automatically be used.

When is Initialization Complete?

- When initialization of dump or trace is complete, you will see '***'

```
 IKJ56650I TIME-10:24:35 AM. CPU-00:00:00 SERVICE-13137 SESSION-02:50:11 MARCH 1,2024
 BLS18122I Initialization in progress for DSNAME('SHARE.S24601.DUMP1')
 BLS18124I TITLE=SHR2ESTA DETECTED ABEND S00C4, REASON 00000011
 BLS18223I Dump written by z/OS 03.01.00-0 SVC dump - level same as IPCS level
 BLS18222I z/Architecture mode system
 BLS18160D May summary dump data be used by dump access?  Enter Y to use, N to bypass.
Y
 BLS18123I 22,776 blocks, 94,748,160 bytes, in DSNAME('SHARE.S24601.DUMP1')
 IKJ56650I TIME-10:24:37 AM. CPU-00:00:00 SERVICE-15128 SESSION-02:50:13 MARCH 1,2024
 BLS18224I Dump of z/OS 03.01.00-0 - level same as IPCS level
 ***
```

- Press ENTER to proceed

Pressing ENTER after '***' will take you to the requested IPCS report/panel.

DIAGNOSING ABENDS AND MESSAGES

ABENDs and Messages

- **System ABENDs:** ABENDxxx ('xxx' is hex completion code)
  - Look up in z/OS MVS System Codes manual, and pursue accordingly
    - SVC dump may be needed
- **User ABENDs:** ABENDUdddd ('dddd' is decimal completion code)
  - Look up in appropriate application manual, and pursue accordingly
    - User dump (e.g. SYSMDUMP) or SVC dump may be needed
- **LOOKAT** can be used to quickly interpret messages
  - https://www.ibm.com/support/pages/node/6855815

The z/OS MVS System Codes manual has details about system ABENDs.

Syntax rule for a return code that goes along with an ABEND code is to document it as: RCxx, where xx is the one-byte (typically) hexadecimal return code. A return code of C would be documented RC0C.

Unfortunately, the terms "reason code" and "return code" have come to be used interchangeably when associated with an ABEND code. Some components issue ABENDs that include the ABEND code, a return code (which further qualifies the ABEND code), and a fullword reason code that further qualifies the return code. The typical syntax for a reason code is RSNxxxxxxxx, where xxxxxxxx is a fullword hexadecimal value.

These syntax rules for ABEND codes, return codes, and reason codes can help when searching for a known issue (e.g. APAR or Technote).

Can I Resolve the Issue?

- Does the ABEND or message provide enough information to resolve?
  - It's a DIY (Do It Yourself) job! ☺
- Or is it unclear from the the ABEND or message what is wrong and how to resolve the issue?
  - Determine who is failing
    - Is this a problem with an in-house application?
    - Is the problem with IBM code?
    - Is OEM vendor code at fault?
- Locate any SVC dumps in problem timeframe
  - Recovery routine may request dump (**recovery dump**) when an ABEND occurs
  - If dump not produced, may need **SLIP dump** of failing address space(s)

Some ABENDs and messages are self-explanatory and may provide enough information for you to resolve the issue yourself. Otherwise, you need to identify who is at fault so the appropriate SME can be engaged.

What is SLIP?

- **SLIP** – **S**erviceability **L**evel **I**ndication **P**rocessing
  - Used to trap ABENDs and messages
  - Primarily software-driven
- **PER** – **P**rogram **E**vent **R**ecording
  - Used to trap hardware events
    - Instruction Fetch
    - Storage Alteration
    - Successful Branch
  - Hardware detects event and invokes SLIP software
  - SLIP software applies filters and takes action(s)
- SLIP/PER is often generically referred to as SLIP

SLIP/PER is the premier tool for trapping programming events in z/OS. It can trap software events such as ABENDs or messages, or it can trap hardware events such as the execution (fetching) of an instruction or the alteration of an area of storage. Its power is in its flexibility both in filtering on event environments and in generating documentation.

The monitoring of hardware events is accomplished by the PER part of SLIP/PER processing. When a monitored hardware event occurs, hardware detects this and signals SLIP software. SLIP software handles the additional filtering and taking of action as appropriate.

SLIP software also provides filtering for the software-generated events and will take action, as appropriate.

SLIP Examples: Collect Dump for ABEND or WTO Message

- SLIP SET,COMP=0C4,JOBNAME=TEST,A=SVCD,
  SDATA=(ALLNUC,CSA,SQA,PSA,LPA,RGN,SUM,TRT),END
  - When an ABEND0C4 occurs in job TEST, collect an SVC dump.
- SLIP SET,COMP=U4038,JOBNAME=PAYROLL,A=SVCD,
  SDATA=(ALLNUC,CSA,SQA,PSA,LPA,RGN,SUM,TRT),END
  - When an ABENDU4038 occurs in job PAYROLL, collect an SVC dump.
- SLIP SET,MSGID=IEA412I,A=SVCD,
  SDATA=(ALLNUC,CSA,SQA,PSA,LPA,RGN,SUM,TRT),END
  - When message IEA421I is issued, collect an SVC dump.

The COMP (shortened to 'C') keyword monitors for a particular completion code (e.g. ABEND code). Wildcarding is allowed: COMP=0CX will monitor for any ABEND0Cx. If a recovery routine converts an ABEND, you need to SLIP on the **original** ABEND code.

The MSGID keyword monitors for a particular message ID up to 10 characters in length. MSGID SLIPs don't work on certain types of messages, most notably branch entry WTOs with the NLCKS, LOADWAIT, or SYNCH=YES parameter specified. MSGID only is applicable for WTO messages to SYSLOG, not messages to other logs. MSGID does accept message IDs that include blanks or special characters if you put single quotes around the message ID. e.g. MSGID='%TEST MSG'. Using quotes also provides a loose form of wildcarding. See the SLIP section in the z/OS System Commands manual for more information.

PSW at Time of Failure/SLIP: IP ST REGS

- Review dump in IPCS to locate PSW associated with failure
  - Issue subcommand: **IP ST REGS**
- PSW at time of dump can help us identify who to engage
  - **PSW address** points to code executing in **PRIMARY ASID**
    - SLIP dump – code where SLIP trap matched
    - Recovery dump – code where failure occurred
  - Need to identify problem code (module CSECT)

```
CPU STATUS:
PSW=07044001 80000000 00000000 2A12D5F4
    (Running in AR, key 0, AMODE 64, DAT ON, SUPERVISOR STATE)
   Disabled for PER
  ASID(X'0011') 2A12D5F4. BPXINPV2+0115F4 IN EXTENDED PRIVATE
  ASID(X'0011') 2A12D5F4. AREA(Subpool252Key00)+5265F4 IN EXTENDED PRIVATE
 ASCB17 at FA5200, JOB(OMVS), for the home ASID
 ASXB17 at 9FD000 and TCB17CA at 9CB518 for the home ASID
 HOME ASID: 0011 PRIMARY ASID: 0011 SECONDARY ASID: 0011
```


The PSW address at the time of the dump points to the code that was executing, which helps you identify who is at fault. In this example, the PSW address at the time of the dump is x’ 2A12D5F4’ in primary ASID(x’11’).

Where Does PSW Point? IP WHERE

- Maps PSW address to specific offset within CSECT or Load Module (collection of CSECTs)
  - WHERE maps local storage address to offset within private load module
  - WHERE maps global storage address as follows:
    - Nucleus: Maps address to offset within CSECT
    - LPA: Maps address to offset within load module
- Example: **IP WHERE 2A12D5F4 ASID(X'11')**

```
**************************************************************************
   ASID(X'0011') 2A12D5F4. BPXINPV2+0115F4 IN EXTENDED PRIVATE
   ASID(X'0011') 2A12D5F4. AREA(Subpool252Key00)+5265F4 IN EXTENDED PRIVATE
**************************************************************************
```


The WHERE command can be used to map a PSW address to a CSECT or load module (collection of CSECTs), plus an offset. For a nucleus address, WHERE resolves directly to the desired CSECT name plus an offset. For an LPA or private address, WHERE resolves to a load module plus an offset.

Browse Storage in Dump to Find CSECT

- If **WHERE** command points to load module, need to browse storage in dump to determine CSECT
  - It is common for program (CSECT) to begin with unconditional branch instruction
    - Opcode/mask x'A7F4xxxx' – BRC (jump)
    - Opcode/mask x'47Fxxxxx' – BC (branch)
  - Branch instruction bypasses module eyecatchers which may include:
    - Module name
    - Maintenance level / compile date
    - Copyright information

- BROWSE Method:
  - '**=1**' on any command line in IPCS takes you to BROWSE menu (IPCS Option 1) that can be used to browse storage
    - Example on next pages will allow us to find CSECT pointed to by failing PSW

If the WHERE command provides you with an LPA or private load module plus an offset, you can attempt to browse storage in the dump to try to identify the CSECT plus an offset. . It is common for a module CSECT to begin with an unconditional branch around module eyecatchers including the module name, maintenance level, and copyright information. For LPA load modules, browsing storage stands less of a chance of being successful because we don't include all of LPA in an SVC dump.

IPCS Option 1 (BROWSE panel) provides you with the means to scroll through storage looking for the starting address of the module CSECT.

An AMBLIST could also be used to map an LPA or private load module plus an offset to a specific CSECT name plus an offset.

IPCS Option 1: BROWSE

- Browsing Storage: IPCS Option 1 (**=1** on any IPCS command line) brings up the following panel. Hit <ENTER> to get to POINTER STACK on next slide.

```
------------------------ IPCS - ENTRY PANEL  ---------------------------------
 CURRENT DEFAULTS:
  Source ==> DSNAME('MVSSPT.S2822.DUMP1A')
  Address space ==> ASID(X'0011')

 OVERRIDE DEFAULTS:                          (defaults used for blank fields)
  Source ==> DSNAME('MVSSPT.S2822.DUMP1A')
  Address space ==>
  Password       ==>

 POINTER:
  Address        ==>                         (blank to display pointer stack)
  Remark         ==>                                        (optional text)
```


An “Address” and “Address space” can be filled in on this screen to jump right to scrollable storage at the specified address. However, hitting <ENTER> on this screen without filling in an “Address” under “POINTER” will bring up a POINTER STACK containing a list of pointers that have been defined for this dump.

IPCS Option 1: BROWSE

```
DSNAME('MVSSPT.S2822.DUMP1A') POINTERS
--------------------------------------
ASID(X'0011') is the default address space
PTR   Address   Address space              Data type
s0001 2A12D5F4  ASID(X'0011')              AREA
      Remarks:
**************** END OF POINTER STACK ************************
```

PF7 – scroll up
PF8 – scroll down

Hit <ENTER> to browse storage at address x'2A12D5F4':

```
ASID(X'0011') ADDRESS(1BE5D6E0.) STORAGE ----------------------------------------------
2A12B890   C3C80000   00000000   A7F40018   00000000   | CH......x4...... |
2A12B8A0   C2D7E7C6   E5D5D340   F0F561F1   F761F1F8   | BPXFVNL 05/17/18 |
2A12B8B0   E4C1F9F6   F2F8F0C4   C2D7E7C9   D5D7E5F2   | UA96280DBPXINPV2 |
------------ (lines omitted) -------------------
2A12D5E0   E380D510   00049102   8045A774   0014E320   | T.N...j...x...T. |
2A12D5F0   D6400017   E3602038   0090A76E   0003A784   | O ..T-....x>..xd |
```

x'2A12D5F4' – x'2A12B898' = x'1D5C' => CSECT name **BPXFVNL+x'1D5C' (UA96280)**

We are browsing storage at address x'2A12D5F4' and scrolling backwards to look for the starting point of a module CSECT. At address x'2A12B898', we see what appears to be the beginning of a module CSECT with an unconditional branch around the module eyecatcher information. The module name is BPXFVNL, which has a maintenance level of UA96280 and a compile date of 05/17/18. Subtracting the CSECT start address x'2A12B898' from the PSW at the time of the dump gives us a CSECT offset of x'1D5C'. Thus, the PSW address points to module CSECT name BPXFVNL+x'1D5C' (UA96280).

Make Note of Key Problem Symptoms (for later search)

ABEND code:

- **ABENDxxx**
  - where ‘xxx’ is system ABEND code in hex
- **ABENDUdddd**
  - where ‘dddd’ is user ABEND code in decimal

Message ID:

- **MSGmmmmmmmm**
  - where ‘mmmmmmmm’ is failing message ID

Failing module CSECT:

- ***CSECT name***
  - extract from ‘WHERE’ command output or browsing storage

Record the key pieces of information you’ve identified thus far (e.g. ABEND code, message ID, module CSECT name). You’ll use this information later to search the IBM Support Portal for known APARs and other technical documentation that may describe your issue.

DIAGNOSING LOOPS AND HANGS

What is a “Hang”?

- Definition: No externally visible work being done by process or function

- Possible triggers:
  - Process is non-dispatchable
    - Requires a resource (e.g. ENQ, Latch) that is not available
    - Waiting for an event that is not occurring
    - May have no work to do
  - Process is dispatchable, but not getting CPU
    - Tuning issue (weight, workload, etc.)
    - Higher priority address space may be looping or consuming excessive CPU
  - Process is looping (High CPU)

There are many ways that an application can appear hung. Similarly, there are many factors which can cause or contribute to a hang.

A looping application may be perceived as hung, since a looping application will not be performing any significant work.

Diagnostic Approach

- Invoke **z/OS Runtime Diagnostics (RTD)**
  - Analyzes system degradation (Sick But Not Dead) issues with broad or system-wide impact, including loops, ENQ/Latch contention, deadlocks
- Take console dump of problem address space(s) before performing recovery actions (e.g. CANCEL)
  - Include any blocker address space(s) associated with ENQ, Latch, and Deadlock events detected by RTD
- If CANCEL does not terminate address space(s), collect another console dump of problem address space(s) prior to FORCE
- If FORCE does not terminate address space(s), collect a console dump of ***MASTER*** address space

If an address space needs to be terminated via CANCEL, it's good practice to take a console dump of the address space prior to the CANCEL, so we can determine what was wrong with the job in the first place. If the CANCEL does not work to terminate the address space, then another dump can be collected so we can determine why the CANCEL processing could not complete. If a FORCE is required to terminate the address space, it's possible that an IPL may ultimately be needed to clean up system resources owned by the job. If a FORCE request hangs, the *MASTER* address space will need to be dumped to diagnose.

z/OS Runtime Diagnostics (RTD)

- Analyzes system, or select address spaces, for problems related to system anomalies
- Performs analysis like very experienced system programmer
  - Faster and more comprehensive – goal of 60 seconds or less
- Invoked via: **MODIFY HZR,ANALYZE,…** **(F HZR,ANALYZE,…)**
  - Recommends next steps
  - Dump can be collected of HZR (RTD) address space and address spaces associated with events via DEBUG parameter (may include up to 15 address spaces in dump)
    - **F HZR,ANALYZE,DEBUG,…**
- Refer to z/OS Problem Management manual for more information about RTD

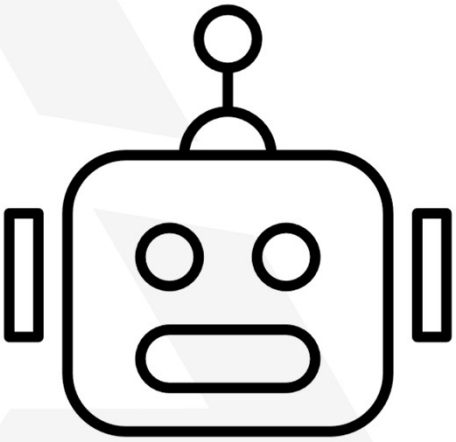
z/OS Runtime Diagnostics drives SBND (Sick But Not Dead) analysis by automating a number of checks and listing next steps for further analysis.

System analysis can be invoked via console command: **F HZR,ANALYZE**. This is similar to analysis of the system performed by an experienced system programmer, but faster.

Output is in the form of a multi-line WTO, with sections identifying anomaly problems/symptoms, location, description of the issue and next step(s) to determine what's wrong.
You can also set up a system data set to capture the output of any F HZR,ANALYZE commands.

z/OS Runtime Diagnostics – Focus Areas

- Critical component anomalies (message analysis)
  - e.g. z/OS UNIX, Console, IOS, SRM, XCF, XES, Logger, Catalog, RRS, WebSphere
- Outstanding WTORs
- JES2 Health exceptions
- Active PER SLIP Trap
- Tight Loop detection
- Local Lock suspension
- ENQ contention (followed to other systems in sysplex)
- GRS Latch contention
  - z/OS UNIX Latch contention
- Deadlock events
- WLM server address space health exceptions



→ **Events are correlated by ASID**

z/OS Runtime Diagnostics Examples – Loop

TCB Recovery Loop

```
F HRZ,ANALYZE
 HZR0200I RUNTIME DIAGNOSTICS RESULT 521
 SUMMARY: SUCCESS
  REQ: 15687 TARGET SYSTEM: SY1    HOME: SY1    2025/05/25 00:47:56
  INTERVAL:  60 MINUTES
  EVENTS FOUND: 1
   PRIORITIES:  HIGH:1  MED:0  LOW:0
   TYPES: LOOP:1
------------------------------------------------------------------------
 EVENT 1: HIGH:LOOP            SYSTEM: SY1     2025/05/25 00:47:57
  ASID: 00CB  JOBNAME: MVTCPIP   TCB: 009CB6D8
  STEPNAME: MVTCPIP   PROCSTEP: TACMDSP4  JOBID: STC97800
  USERID: MVTCPIP   JOBSTART: 2025/05/12 18:52:47
 ERROR : THE ADDRESS SPACE MIGHT BE IN A TCB RECOVERY LOOP.
 ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.
------------------------------------------------------------------------
```

TCB-Enabled Loop

```
F HZR,ANALYZE
 HZR0200I RUNTIME DIAGNOSTICS RESULT
 SUMMARY: SUCCESS
  REQ: 1 TARGET SYSTEM: SY1       HOME: SY1      2025/08/27 08:23:01
  INTERVAL:  60 MINUTES
  EVENTS FOUND: 1
   PRIORITIES:  HIGH:1  MED:0  LOW:0
   TYPES: LOOP:1
------------------------------------------------------------------------
 EVENT 1: HIGH:LOOP             SYSTEM: SY1      2025/08/27 08:23:07
  ASID: 0029  JOBNAME: RTDLP     TCB: 008FE990
  STEPNAME: STEP1     PROCSTEP:           JOBID:JOB00041
  USERID: WELLIE0   JOBSTART: 2025/08/27 08:17:50
 ERROR : ADDRESS SPACE MIGHT BE IN A TCB-ENABLED LOOP.
 ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.
------------------------------------------------------------------------
```


The RTD output in the green box on the left shows a TCB recovery loop with TCB(x'9DB6D8') in ASID(x'CB') MVTCPIP.

The RTD output in the purple box on the right shows a TCB-enabled loop with TCB(x'8FE990') in ASID(x'29') RTDLP.

Consider issuing **F HZR,ANALYZE,DEBUG=LOOP** to collect a dump of address spaces that are associated with each LOOP event.

z/OS Runtime Diagnostics Examples –Contention

ENQ Contention

```
F HZR,ANALYZE
 HZR0200I RUNTIME DIAGNOSTICS RESULT
 SUMMARY: SUCCESS
  REQ: 1 TARGET SYSTEM: SY1      HOME: SY1       2025/09/11
08:23:00
  INTERVAL:  60 MINUTES
  EVENTS FOUND: 1
   PRIORITIES:  HIGH:1  MED:0  LOW:0
   TYPES: ENQ:1
------------------------------------------------------------------
 EVENT 1: HIGH:ENQ               SYSTEM: SY1     2025/09/11
08:23:00
  QNAME: SYSZMCS    SCOPE: SYSTEMS
  RNAME: SYSMCS#MCS
               ASID  JOB NAME  TCB/WEB    SYSTEM    WAIT TIME
  TOP WAITER : 002A  FENQ2     008C9BE0   SY1       00:05:05
  TOP BLOCKER: 002A  FENQ2     008C9E00   SY1
  OTHER WAITERS FOR THIS RESOURCE:
       WAITER: 0029  FENQ3     008FE990   SY2       00:02:13
       WAITER: 003B  FENQ5     008C5540   SY1       00:02:08
 ERROR : ADDRESS SPACES MIGHT BE IN ENQ CONTENTION.
 ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE BLOCKING JOBS
 ACTION: AND ASIDS.
------------------------------------------------------------------
```

Latch Contention and z/OS UNIX Latch Contention

```
F HZR,ANALYZE
 HZR0200I RUNTIME DIAGNOSTICS RESULT
 SUMMARY: SUCCESS
  REQ: 1 TARGET SYSTEM: SY1      HOME: SY1       2025/06/02 10:49:30
  INTERVAL:  60 MINUTES
 EVENTS FOUND: 2
   PRIORITIES:  HIGH:2  MED:0  LOW:0
   TYPES: LATCH:1  OMVS:1
------------------------------------------------------------------
 EVENT 1: HIGH:LATCH             SYSTEM: SY1     2025/06/02 10:49:30
  LATCH SET NAME: SYSTEST.LATCH_TESTSET
  LATCH NUMBER: 3          CASID: 0034  CJOBNAME: TSTLATC0
               ASID  JOB NAME  TCB/WEB    SYSTEM    WAIT TIME
  TOP WAITER : 0034  TSTLATC0  008F8368   SY1       00:05:14
  TOP BLOCKER: 0034  TSTLATC0  008F8588   SY1
 ERROR : ADDRESS SPACES MIGHT BE IN LATCH CONTENTION.
 ACTION: D GRS,AN,LATCH,DEP,CASID=0034,LAT=(SYSTEST.L*,3),DET
 ACTION: TO ANALYZE THE LATCH DEPENDENCIES. USE YOUR SOFTWARE MONITORS
 ACTION: TO INVESTIGATE BLOCKING JOBS AND ASIDS.
------------------------------------------------------------------
 EVENT 2: HIGH:OMVS              SYSTEM: SY1     2025/06/02 10:49:30
  ASID: 000E  JOBNAME: OMVS
  MOUNT LATCH WAITERS: 1
  FILE SYSTEM LATCH WAITERS: 0
  XSYS AND OTHER THREADS WAITING FOR z/OS UNIX: 1
 ERROR : z/OS UNIX MIGHT HAVE FILE SYSTEM LATCH CONTENTION.
 ACTION: D OMVS,W,A TO INVESTIGATE z/OS UNIX FILE SYSTEM LATCH
 ACTION: CONTENTION, ACTIVITY AND WAITING THREADS. USE YOUR SOFTWARE
 ACTION: MONITORS TO INVESTIGATE BLOCKING JOBS AND ASIDS.
------------------------------------------------------------------
```


The RTD output in the green box on the left shows ENQ contention for a SYSTEMS-scoped ENQ with QNAME=SYSZMCS and RNAME=SYSMCS#MCS. The top blocker is TCB(x'8C9E00') in ASID(x'2A') FENQ2 running on SY1.

Consider issuing **F HZR,ANALYZE,DEBUG=ENQ** to collect a dump of address spaces that are associated with each ENQ event.

The RTD output in the purple box on the right shows 2 instance of latch contention.

The first event shows contention with Latch#3 in Latch Set=SYSTEST.LATCH_TESTSET. The top blocker is TCB(x'8F8588') in ASID(x'34') TSTLATC0.

The second event shows contention with the z/OS UNIX Mount Latch and a long-waiting z/OS UNIX thread. The 'D OMVS,W,A' output can help shed more light on what is causing the contention/long wait.

Consider issuing **F HZR,ANALYZE,DEBUG=(LATCH,OMVS)** to collect a dump of address spaces that are associated with each OMVS and LATCH event.

z/OS Runtime Diagnostics Examples – Deadlock

SHARE
EDUCATE • NETWORK • INFLUENCE

Deadlock

```
 EVENT 1: HIGH:DEADLOCK      SYSTEM: SY1      2025/04/014 08:49:10
RESOURCE: ENQ
 QNAME: SYSZDLK    SCOPE: SYSPLEX
 RNAME: DLKEN32
              ASID  JOB NAME  TCB/WEB   SYSTEM    WAIT TIME
 TOP WAITER : 0029  DLKSTART  008C5A60  SY1       00:02:23
 BLOCKER    : 0029  DLKSTART  008C5CF0  SY1
 OTHER WAITERS FOR THIS RESOURCE:
      WAITER: 0029  DLKSTART  008FE990  SY1       00:02:13
      WAITER: 0029  DLKSTART  008C5540  SY1       00:02:08
      WAITER: 0029  DLKSTART  008C57D0  SY1       00:02:08
       ------------------------------------------------------
RESOURCE: ENQ
 QNAME: SYSZDLK    SCOPE: SYSPLEX
 RNAME: DLKEN31
              ASID  JOB NAME  TCB/WEB   SYSTEM    WAIT TIME
 TOP WAITER : 0029  DLKSTART  008C5CF0  SY1       00:02:23
 BLOCKER    : 003A  DLKSTART  008C5E88  SY2
       ------------------------------------------------------
RESOURCE: ENQ
 QNAME: SYSZDLK    SCOPE: SYSPLEX
 RNAME: DLKEN33
              ASID  JOB NAME  TCB/WEB   SYSTEM    WAIT TIME
 TOP WAITER : 003A  DLKSTART  008C5E88  SY2       00:02:23
 TOP BLOCKER: 0029  DLKSTART  008C5A60  SY1
ERROR : ADDRESS SPACES WERE DEADLOCKED AT THE TIME OF THE ANALYZE
ERROR : REQUEST.
ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE BLOCKERS TO
ACTION: DETERMINE IF THE DEADLOCK STILL EXISTS AND TAKE
ACTION: APPROPRIATE ACTION TO ELIMINATE THE DEADLOCK.
```

Deadlock

```
EVENT 2: HIGH:DEADLOCK          SYSTEM: SY1      2025/03/01 09:01:40
RESOURCE: ENQ
 QNAME: SYSZDLK    SCOPE: STEP
 RNAME: DLKCMB1
              ASID  JOB NAME  TCB/WEB   SYSTEM    WAIT TIME
 TOP WAITER : 0029  DLKSTART  008C5E88  SY1       00:00:49
 BLOCKER    : 0029  DLKSTART  008C5CF0  SY1
       ------------------------------------------------------
RESOURCE: LATCH
 LATCH SET NAME: SYSRTD.DLKCMB1TEST.LATCHSET
 LATCH NUMBER: 0          CASID: 0029  CJOBNAME: DLKSTART
              ASID  JOB NAME  TCB/WEB   SYSTEM    WAIT TIME
 TOP WAITER : 0029  DLKSTART  008C5CF0  SY1       00:00:49
 TOP BLOCKER: 0029  DLKSTART  008C5E88  SY1
 OTHER WAITERS FOR THIS RESOURCE:
      WAITER: 0029  DLKSTART  008FE990  SY1       00:00:33
ERROR : ADDRESS SPACES WERE DEADLOCKED AT THE TIME OF THE ANALYZE
ERROR : REQUEST.
ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE BLOCKERS TO
ACTION: DETERMINE IF THE DEADLOCK STILL EXISTS AND TAKE APPROPRIATE
ACTION: ACTION TO ELIMINATE THE DEADLOCK.
-------------------------------------------------------------------
```


The RTD output in the green box on the left shows a multi-system (SY1 and SY2) deadlock between 3 DLKSTART tasks involving 3 ENQs.

The RTD output in the purple box on the right shows a deadlock between 2 tasks in a DLKSTART address space involving an ENQ and a LATCH.

Consider issuing **F HZR,ANALYZE,DEBUG=DEADLOCK** to collect a dump of address spaces that are associated with each DEADLOCK event.

Requesting Console Dump

- Use z/OS **DUMP** command to collect data for analysis of persistent problem (e.g. high CPU, loop, hang)
- Console dump can be initiated from an authorized console, SDSF, or other command-supporting utility
  - **DUMP {COMM={(title)} [,PARMLIB=xx]}**
- Either reply to WTOR or specify PARMLIB member to be used (**IEADMCxx**).
  - IEADMCxx members can be concatenated with PARMLIB=(xx,yy)
    - **DUMP COMM=(dump title),PARMLIB=(12,US)**

BROWSE SYS1.PARMLIB(IEADMC12)
Command ===>
********** Top of Data **********
ASID=(3,000A),
SDATA=(CSA,SQA,LPA,RGN,TRT,PSA,GRSQ,ALLNUC,SUM),
DSPNAME=('RASP'.*)
********** Bottom of Data **********

BROWSE SYS1.PARMLIB(IEADMCUS)
Command ===>
********** Top of Data **********
JOBNAME=(OMVS,ZFS),DSPNAME=('OMVS'.SYS*,'OMVS'.BPX*),
REMOTE=(SYSLIST=*('OMVS','ZFS'),SDATA,DSPNAME)

Refer to z/OS MVS Initialization and Tuning Reference for more details about IEADMCxx member

Console Dump – What to Dump

- SDATA parm identifies storage areas to be included in SVC dump.
  - SDATA=(ALLNUC,CSA,SQA,LPA,PSA,SWA,RGN,TRT,CSA,.....)
- Dumping Address Spaces and Dataspaces
  - ASID=(3B,6F,E,27,A1,....) <= [hex values]
  - JOBNAME=(OMVS,GRS,…)
  - DSPNAME=('jobname'.dspname)
- Additional dump options available in a sysplex environment
  - SDATA=(WLM,COUPLE,XESDATA,.....)
  - STRLIST=(strname=, lockentries=, listnum=...)
  - REMOTE=

What to dump depends upon the problem encountered (e.g. hung job, resource contention, component-specific failure, etc.).

Refer to z/OS MVS System Commands (DUMP Command) for more details about the parameters for a console dump.

Make Note of Key Problem Symptoms (for later search)

Loop or High CPU:

- LOOP
- HIGH CPU
- *CSECT name/load module associated with Loop/High CPU*
  - extract from software monitor

Hang:

- HANG
- DEADLOCK
- *Latch Set name*
- LATCH#nn
  - where 'nn' is decimal latch number without leading zeros
- ENQ
- *ENQ QNAME (Queue major name) / RNAME (Resource minor name)*

Record the key pieces of information you've identified thus far.

For a loop/high CPU, be sure to include the keywords LOOP or HIGH CPU, as well as the name of the CSECT or load module associated with the loop/high CPU, which may be extracted from a software monitor.

For a hang, be sure to include the keywords HANG or DEADLOCK. If an ENQ is involved, include the keyword ENQ and the QNAME/RNAME of the ENQ, and if a latch is involved, include the Latch Set name and the LATCH#nn keyword.

You'll use this information later to search the IBM Support Portal for known APARs and other technical documentation that may describe your issue.

GATHER OTHER RELATED INFORMATION

System Log (SYSLOG)

- Provides message history from single system leading up to problem
- Referred to as the hard-copy log
- The system log consists of:
  - Any messages routed to the system log from any system component or program
  - All messages issued through WTL macros
  - All messages entered by operator LOG commands

The SYSLOG (hard-copy log) provides a history of messages and commands from a single system. Refer to the Introduction pages of the z/OS MVS System Messages manual for more details about the format of SYSLOG messages for both a JES2 and JES3 system.

The SYSLOG is actually a SYSOUT data set that stores the messages and commands from the current system. SYSOUT data sets are output spool data sets on Direct Access Storage Devices (DASD) provided by the Job Entry Subsystem (either JES2 or JES3).

Operations Log (OPERLOG)

- Combined SYSLOG data (hard-copy log) from all systems in sysplex
- OPERLOG consists of the following data:
  - Messages to and from all consoles
  - Commands and replies entered by the operator
- Use OPERLOG rather than SYSLOG when debugging a problem involving multiple systems in sysplex

OPERLOG facilitates the debugger's need to coordinate messages and events across multiple systems in a sysplex. OPERLOG requires the use of z/OS MVS Logger.

LOGREC

- Provides history of hardware/software error events from single system leading up to problem, including:
  - ABEND records
  - Symptom Records
- Run IFCEREP1 (EREP) against SYS1.LOGREC dataset to obtain LOGREC records
  - Refer to speaker notes for sample JCL

Refer to z/OS MVS Diagnosis: Tools and Service Aids for more information about LOGREC records.

The following example shows how to generate detail edits and summaries of all software and operational records:

```
//STEP7 EXEC PGM=IFCEREP1,PARM='CARD'
//ACCIN DD DSN=EHISTORY,DISP=(OLD,PASS)
//DIRECTWK DD UNIT=SYSDA,
// SPACE=(CYL,5,,CONTIG)
//EREPPT DD SYSOUT=A,DCB=BLKSIZE=133
//TOURIST DD SYSOUT=A,DCB=BLKSIZE=133
//SYSIN DD DSN=EREP.PARMS(STEP7),
// DISP=(OLD,PASS)
PRINT=PS
TYPE=SIE
HIST
ACC=N
ENDPARM
```

LOGREC Log Stream

- Combined LOGREC data from all systems in sysplex
- LOGREC log stream consists of following data across sysplex:
  - Hardware failures
  - Software errors
  - Symptom records
- Use LOGREC log stream, rather than LOGREC data set, when debugging multi-system problem
- Run IFCEREP1 (EREP) against log stream SYSPLEX.LOGREC.ALLRECS to obtain LOGREC records
  - Refer to speaker notes for sample JCL

Use the following JCL to produce an event history report from records on the LOGREC log stream. By specifying a print data set, EREPPT, the report can be browsed online for an overview of significant activity.

```
//EREPNOW EXEC PGM=IFCEREP1,REGION=4M,
// PARM='CARD'
//ACCIN DD DSN=SYSPLEX.LOGREC.ALLRECS,
// DISP=SHR,
// SUBSYS=(LOGR,IFBSEXIT,,)
//DIRECTWK DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//EREPPT DD DSN=EREP.EVENT,DISP=(NEW,CATLG),
// DCB=BLKSIZE=133,
// UNIT=SYSDA,SPACE=(CYL,(25,5))
//TOURIST DD SYSOUT=A,DCB=BLKSIZE=133
//SYSABEND DD SYSOUT=A
//SYSIN DD *
PRINT=PS
EVENT
HIST
ACC=N
TYPE=SIE
ENDPARM
```

Navigating LOGREC Data

Useful elements:

- JOBNAME: identifies failing jobname
- ERRORID: contains failing ASID and time of error
- TIME OF ERROR INFORMATION: provides failing PSW
- RECOVERY ROUTINE ACTION: indicates if dump was requested

F 'WARE REC'

- Scroll through SOFTWARE RECORDs

LOGREC data is useful for reviewing the most recent ABENDs that occurred prior to the dump. The SOFTWARE RECORDs are often of most value.

LOGREC: Example

```
TYPE:   SOFTWARE RECORD       REPORT:   SOFTWARE EDIT REPORT          DAY.YEAR
        (PROGRAM INTERRUPT)                                REPORT DATE: 066.24
FORMATTED BY: IEAVTFDE  HBB77E0                             ERROR DATE: 057.24
                              MODEL:    2828                            HH:MM:SS.TH
                              SERIAL:   011011                    TIME: 17:14:50.35

JOBNAME: INIT        SYSTEM NAME: SYS1
ERRORID: SEQ=55368   CPU=0000   ASID=0063   TIME=17:14:50.3
OTHER SERVICEABILITY INFORMATION

  RECOVERY ROUTINE LABEL:   STAEEXIT
  SUBFUNCTION:              SMF TERMINATION EXIT
```


ASID: ASID that encountered the error
TIME: Time the error occurred

This error occurred in ASID(x'63') INIT at 17:14:50.3 on Julian day 57 in the year 2024 (Feb 26, 2024).

LOGREC: Example (cont)

```
SEARCH ARGUMENT ABSTRACT

  PIDS/5752SC100 RIDS/IEFACTRT#L RIDS/IEFACTRT AB/S00C4 PRCS/00000010
  RIDS/IEFTB721#R

  SYMPTOM             DESCRIPTION
  -------             -----------
  PIDS/5752SC100      PROGRAM ID: 5752SC100
  RIDS/IEFACTRT#L     LOAD MODULE NAME: IEFACTRT
  RIDS/IEFACTRT       CSECT NAME: IEFACTRT
  AB/S00C4            SYSTEM ABEND CODE: 00C4
  PRCS/00000010       ABEND REASON CODE: 00000010
  RIDS/IEFTB721#R     RECOVERY ROUTINE CSECT NAME: IEFTB721
```


The error was an ABEND0C4 RSN10 in module CSECT IEFACTRT.

LOGREC: Example (cont)

```
TIME OF ERROR INFORMATION

 PSW: 07042000 80000000 00000000 0A485A9E
 INSTRUCTION LENGTH: 04    INTERRUPT CODE: 0010
 FAILING INSTRUCTION TEXT: 07FE50ED 000C5821 00005822
 TRANSLATION EXCEPTION ADDRESS: 00000000_4B106800

 BREAKING EVENT ADDRESS: 00000000_0A485A8C
 REGISTERS 0-7
 GR: 00011000 4B106491 27200DD0 00000000  00000004 00000000 00000000 00000000
 AR: 00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
 REGISTERS 8-15
 GR: 00000000 00000000 00000000 00000000  00000000 009E6778 00FD6FE8 00000000
 AR: 00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000

 HOME ASID: 0063     PRIMARY ASID: 0063     SECONDARY ASID: 0063
 PKM: 80C0           AX: 0000               EAX: 0000

 THIS TASK'S ASID/TCB: 0063/009F8588
```


The failing PSW address is x'0A485A9E'.

LOGREC: Example (cont)

```
RECOVERY ENVIRONMENT

  RECOVERY ROUTINE TYPE: ESTAE RECOVERY ROUTINE
  RECOVERY ROUTINE ENTRY POINT: 0B4E9446
  A PREVIOUS RECOVERY EXIT FAILED.
  THE PREVIOUS RECOVERY EXIT PERCOLATED TO THIS ONE.
  THE RB ASSOCIATED WITH THIS EXIT WAS NOT IN CONTROL AT THE TIME OF ERROR.
  USER REQUESTED NO I/O PROCESSING.

RECOVERY ROUTINE ACTION

  THE RECOVERY ROUTINE RETRIED TO ADDRESS 0B4E98B2.
  AN SVC DUMP WAS NOT REQUESTED.
  NO LOCKS WERE REQUESTED TO BE FREED.
  THE SDWA WAS REQUESTED TO BE FREED BEFORE RETRY.
```


An SVC dump was not requested by the recovery routine for this ABEND0C4 RSN10 in module IEFACTRT.

Make Note of Any Related ABENDs or Messages (for search)

ABEND code:

- **ABENDxxx**
  - where ‘xxx’ is system ABEND code in hex
- **ABENDUdddd**
  - where ‘dddd’ is user ABEND code in decimal

Message ID:

- **MSGmmmmmmmm**
  - where ‘mmmmmmmm’ is failing message ID

Record any additional ABENDs or messages leading up to the problem that may be related. You’ll use this information to search the IBM Support Portal for known APARs and other technical documentation that may describe your issue.

SEARCH FOR KNOWN ISSUE

Search for Known Issue

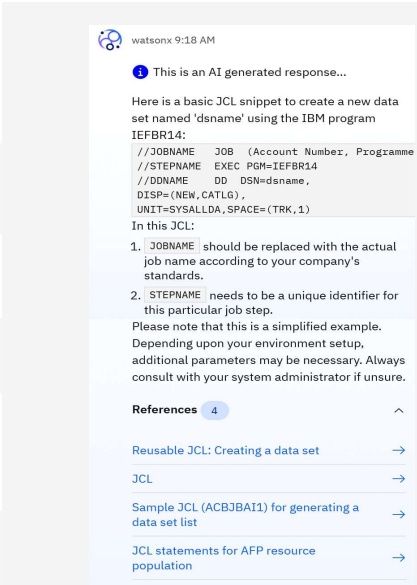
- Before reaching out to IBM, search for known APARs and technical documentation
  - https://www.ibm.com/mysupport
- Construct search with standardized keywords for all relevant pieces of information you have gathered to this point
  - ABEND or Message: ABENDxxx, ABENDUdddd, MSGmmmmmmmm, *CSECT name*
  - Loop or High CPU: LOOP, HIGH CPU, *CSECT or load module name*
  - Hang: HANG, DEADLOCK, ENQ, *ENQ QNAME/RNAME*, Latch#nn, *Latch Set name*
- Refer to **INFO APAR II07428** for list of common keywords

Once you've identified the key pieces of diagnostic information, you can try searching the IBM Support Portal for known APARs and other technical documentation that may describe your issue. To maximize your chances of finding a relevant APAR/document, you should try to use standardized keywords in your search, which are documented in INFO APAR II07428.

Elevate your support experience with watsonx

Get answers, technical guidance, and case support with just a few clicks.

See it in action.
The example below shows the response of the assistant when a user asks it to "Provide JCL to create a dataset in zOS"



Virtual assistant can speed up your IT support experience.
The virtual assistant is available 24x7 to answer common technical support questions.

How to get started:

- Visit the IBM Support site: www.ibm.com/mysupport
- Look for the blue icon in the bottom right corner.
- Select "Ask a Product Question" and log in with your IBM id.
- Start getting the support you need. Provide feedback in the assistant.

SUMMARY

Summary

- IPCS is the tool that lets us investigate dumps.
- ABENDs and messages may be self-explanatory, but sometimes a dump is required to further diagnose.
- z/OS Runtime Diagnostics (RTD) is a useful tool for detecting and helping to diagnose loops and hangs.
- The most recent message history and error history may give us clues about what happened leading up to a failure.

THANK YOU!

John Shebey jshebey@us.ibm.com

Experience more with IBM

Visit us at the IBM Booth #113

After a full day of technical sessions, take a break with us!

Connect with our experts, snap a photo with the z17 Plexi or the latest Telum II, and get an up-close look at our Spyre Accelerator.

Come back each day for fresh topics and demos at our expert stations.

Think 2026 (May 4-7) - Boston

Join 5000+ senior business and technology leaders who are seizing the AI revolution to unlock unprecedented growth and productivity at Think 2026.

Find out more information using the QR code below.

IBM Digital Asset Haven

IBM Digital Asset Haven is the operational backbone for financial institutions and regulated enterprises entering the digital asset economy.

Find out more information using the QR code below.

Your feedback is important!

SHARE
EDUCATE • NETWORK • INFLUENCE

Submit a session evaluation for each session you attend:

www.share.org/evaluation