



IBM Systems

SDSF: What's New in z/OS 1.12?

Chip Wood
IBM Poughkeepsie
chipwood@us.ibm.com

© 2010 IBM Corporation 1

In z/OS 1.12 SDSF, several new functions are introduced, including additional JES3 support, a Health Checker History display, and support for Java.

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

IBM®
MVS
JES2
JES3
RACF®
REXX
z/OS®
zSeries®

* Registered trademarks of IBM Corporation


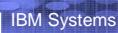
The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
UNIX is a registered trademark of The Open Group in the United States and other countries.
SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.



z/OS V1R12 SDSF Release Highlights:

- **Health Checker History panel**
- **Enhancements to INIT panel**
- **Additional JES3 support (PR panel)**
- **Search help support**
- **REXX log support**
- **SDSF data available through JAVA**

© 2010 IBM Corporation


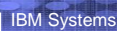


Installation

- z/OS V1R12 SDSF packaging is similar to R11:
 - SDSF base FMID HQX7770
 - Contains common code and JES3 support
 - SDSF JES2 feature FMID JJE777S
 - Contains JES2 support
 - JES2 installations must install both HQX7770 and JJE777S
 - By default no assemblies are done at SMP/E APPLY time

© 2010 IBM Corporation

4





Migration & Coexistence Considerations

- Sharing SDSF 1.12 Server Params with lower releases of SDSF
 - For this function, if you are sharing ISFPRMxx with SDSF 1.10 and/or 1.11 systems you must install the toleration PTFs associated with APAR PK86390 :
 - For SDSF 1.10, you must install PTF UK90017 and if running SDSF with JES2 you must also install PTF UK90022
 - For SDSF 1.11 you must install PTF UK90018 and if running SDSF with JES2 you must also install PTF UK90023
- Displaying printers from a JES2 1.12 system on lower releases
 - Same APARs required to correctly display **Status** and **Work-Selection** columns.
 - JES2 compatibility APAR(s) in z/OS 1.11 data gathering code, when in a mixed MAS with z/OS 1.11 and z/OS 1.12
 - OA31703 and all prerequisites


© 2010 IBM Corporation

5



Health Checker History: Overview

- **Problem Statement/Need Addressed :**
The IBM Health Checker provides an option for saving Health Checks to a logstream for historical purposes. SDSF could only present the user with the current status of health checks via the SDSF Health Checker display.
- **Solution:**
Provide a new action on the SDSF Health Checker display to enable the user to view the history of a health check
- **Benefit:**
Users can now easily view the history and details of health check runs



The screenshot shows the SDSF Health Checker History display. At the top, there is a blue header with 'IBM Systems' and the IBM logo. Below the header, the title 'Health Checker History' is displayed in blue. A bullet point explains the new LogStream column. The main display area shows a terminal window with a menu bar (File, Edit, View, Communication, Actions, Window, Help) and a list of health checker items. A red arrow points to the 'LogStream' column, which is highlighted in yellow. The items listed are:

NP	NAME	LogStream
	ASM_LOCAL_SLOT_USAGE	HZS.HEALTH.CHECKER.HISTORY
	ASM_NUMBER_LOCAL_DATASETS	HZS.HEALTH.CHECKER.HISTORY
	ASM_PAGE_ADD	HZS.HEALTH.CHECKER.HISTORY

© 2010 IBM Corporation

The screenshot shows the SDSF Health Checker History display. At the top, there is a blue header with 'IBM Systems' and the IBM logo. Below the header, the title 'Health Checker History' is displayed. A bulleted list explains that the support is invoked by the new L (ListHistory) action on the SDSF Health Checker display, which is valid for checks with a logstream associated to them.

The main part of the screenshot is a terminal window titled 'TS01 - [43 x 80]'. It shows the SDSF Health Checker History display for SY1. The display has a menu bar with 'Display', 'Filter', 'View', 'Print', 'Options', 'Search', and 'Help'. The main content is a table with columns for 'NP', 'NAME', 'CheckOwner', and 'State'. The first row is highlighted with a red 'L' in the 'NP' column, indicating the selected check.

NP	NAME	CheckOwner	State
L	ASM_LOCAL_SLOT_USAGE	IBMASM	ACTIVE(ENABLED)
	ASM_NUMBER_LOCAL_DATASETS	IBMASM	ACTIVE(ENABLED)
	ASM_PAGE_ADD	IBMASM	ACTIVE(ENABLED)
	ASM_PLPA_COMMON_SIZE	IBMASM	ACTIVE(ENABLED)
	ASM_PLPA_COMMON_USAGE	IBMASM	ACTIVE(ENABLED)
	CATALOG_IMBED_REPLICATE	IBMCATALOG	ACTIVE(ENABLED)
	CEE_USING_LE_PARMLIB	IBMCEE	ACTIVE(ENABLED)
	CNZ_AMRF_EVENTUAL_ACTION_MSGS	IBMCNZ	ACTIVE(ENABLED)
	CNZ_CONSOLE_MASTERAUTH_CMDSYS	IBMCNZ	ACTIVE(ENABLED)
	CNZ_CONSOLE_MSCOPE_AND_ROUTCODE	IBMCNZ	ACTIVE(ENABLED)
	CNZ_CONSOLE_ROUTCODE_11	IBMCNZ	ACTIVE(ENABLED)
	CNZ_EMCS_HARDCOPY_MSCOPE	IBMCNZ	ACTIVE(ENABLED)
	CNZ_EMCS_INACTIVE_CONSOLES	IBMCNZ	ACTIVE(ENABLED)

At the bottom of the terminal window, there is a copyright notice: '© 2010 IBM Corporation'.

In this example the L action invokes the SDSF Health Checker History display for the ASM_LOCAL_SLOT_USAGE check

IBM Systems

Health Checker History


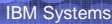
- The L action displays the new CKH panel

```

TS01 - [43 x 80]
File Edit View Communication Actions Window Help
Display Filter View Print Options Search Help
-----
SDSF CK HISTORY  ASM_LOCAL_SLOT_USAGE                LINE 1-2 (2)
COMMAND INPUT ===>                                SCROLL ===> HALF
ACTION=//-Block,=-Repeat,+-Extend,S-Browse,SB-ISPFBrowse,SE-ISPFEdit,X-Print,
ACTION=XC-PrintClose,XD-PrintDS,XDC-PrintDSClose,XF-PrintFile,
ACTION=XFC-PrintFileClose,XS-PrintSysout,XSC-PrintSysoutClose,
NP      COUNT  CheckOwner      Status      Result  Diag1    Diag2
  2      2      IBMASM          EXCEPTION-MEDIUM  8  00000000 00000000
  1      1      IBMASM          EXCEPTION-MEDIUM  8  00000000 00000000
    
```

© 2010 IBM Corporation


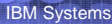
In this example the L action invokes the SDSF Health Checker History display for the ASM_LOCAL_SLOT_USAGE check



Health Checker History


- From the Health Checker History display the user can:
 - Browse a specific run of a check
 - Print a specific run of a check
- The information provided is for each run of a check and includes:
 - Check owner
 - Status
 - Result
 - Diagnostic data
 - Start and stop time
 - System and sysplex name
 - Check name which is also seen on the panel title line


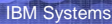
© 2010 IBM Corporation



Health Checker History


- By default, SDSF will collect the last 10 iterations of a check. The user can override this default using the new SET CKLIM command.
 - The minimum number of checks can be 1 and the maximum is 999,999.
- You can also override the SDSF default of 10 iterations using the new Panel.CK.DefaultCKLim custom property.
 - The user can override this default via the SET CKLIM command





INIT panel changes

- The INIT panel now uses SSI 82 to obtain initiator information
 - JES2 can present information about initiators on all MAS members across the SSI, without SDSF Sysplex Communication active, if all MAS members are at z/OS 1.12 or higher
 - WLM initiators can be displayed
 - **INIT ALL** – displays all initiators
 - **INIT JES** – displays JES initiators
 - **INIT WLM** – displays WLM initiators
 - Default is ALL
 - **Command.INIT.DefaultJESManaged** custom property can be specified to make default JES, as in prior releases



12

INIT panel changes – New columns

New columns:

SrvClass	Service class of the active job (JES) or service class the initiator is running (WLM)
Mode	Initiator Mode (JES or WLM)
Group	Group name (JES2 or WLM)
ResType	Resource Type (INIT)

IBM Systems IBM

INIT Panel - example


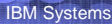
ID	Status	Mode	Classes	JobName	Stepname	ProcStep	JobID	C	ASID
1	INACTIVE	JES	A						39
2	ACTIVE	JES	AB	MANYSPN2	STEP1	G	J0000022	B	43
3	DRAINED	JES	ABC						
4	DRAINED	JES	ABC						
5	DRAINED	JES	JX						
6	DRAINED	JES	A						
7	DRAINED	JES	A						
8	DRAINED	JES	A						
9	DRAINED	JES	JX						
10	DRAINED	JES	A						
	WLM ACTIVE	WLM		MANYSPIN	STEP1	G	J0000021	A	38
	WLM ACTIVE	WLM		MANYSPIN	STEP1	G	J0000024	A	44

*SDSF

MA g 04/021

© 2010 IBM Corporation

Here's an example of the INIT panel with a couple of active WLM inits. Note the new rows, displaying the WLM inits, and the new "Mode" column.


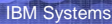


PR panel changes

- The PR panel now uses SSI 83 to obtain printer information
 - JES2 can present information about printers on all members across the SSI, without SDSF Sysplex Communication active, if all MAS members are at z/OS 1.11 or higher
 - Printers can now be displayed under JES3.
 - New columns and overtypes for both JES2 and JES3
 - See Appendix for full list of changes

© 2010 IBM Corporation

15





PR panel – JES2

- Invoked via **PR** command
- Printers on all MAS members are accessible via **SYSNAME** command, even if SDSF Sysplex Communication is not active
 - All members must be a z11 or higher
- Fixed field (device name) increased to 10 characters
 - Rnnnnn.PRn for high remotes rather than RnnnnnPn
 - Custom property to revert to 8-character fixed field
Panel.PR.DevnameAlwaysShort
- New columns and overtypes (see Appendix)
- Backspace/Forwardspace action characters and SET ACTION display now match
 - May specify either **BCn** or **BC,n**

© 2010 IBM Corporation

16





PR panel – JES3

- Invoked via **PR** command
- Currently displays local printers only
 - **LCL/RMT** options on PR command disallowed
 - Device number ranges disallowed
- **Sysname** column always points to global
 - **FSASysNm** is system where FSS printer is running if on a local
- Many overtypeable characteristics can be specified on *START, *RESTART, or *CALL,WTR commands
 - Overtypes of these characteristics require an action character of **E**, **S**, or **X**
 - “**ACTION REQUIRED**” error message if not specified

© 2010 IBM Corporation

17



PR panel – JES3 (continued)

- **E** (restart), **S** (start), and **X** (call) can be qualified by multiple additional characters
 - For example “**EADHR**” results in
 ***R device,A,D,HOLD,RSCD**
 - Action qualifiers optionally separated by commas (**EA,D,H,R**)
 - More combinations/permutations than can be listed on **SET ACTION**
 - “**S+ADMTX-StartOptions**” on **SET ACTION LONG**
 - **S,SA,SD,SM,ST,SX** on **SET ACTION SHORT**

© 2010 IBM Corporation

18

JES3 Printer Actions

```

SDSF PRINTER DISPLAY SY1                                LINE 1-10 (21)
COMMAND INPUT ==> set action long                       SCROLL ==> CSR
PREFIX=* DEST=(ALL) OWNER=* SYSNAME=
ACTION=// -Block, -Repeat, + -Extend, BC-BackCkpt, BCnP-BackNumCkpt, BD-BackTop,
ACTION=BN-BackCkptN, BNnP-BackNumCkptN, C-Cancel, CG-CancelGroup, CJ-CancelJob,
ACTION=CP-CancelPosition, CI-CancelStop, D-Display, DL-DisplayLong,
ACTION=EH-ADHJLMRTX-RestartOptions, E-Restart, EH-RestartHold, EJ-RestartJob,
ACTION=ER-RestartRescan, FN-ForwardNum, FC-ForwardCkpt, FCnP-ForwardNumCkpt,
ACTION=FN-ForwardCkptN, FNnP-ForwardNumCkptN, K-ForceFSS, L-Fail, LD-FailDump,
ACTION=S+ADMTX-StartOptions, S-Start, V-VaryOn, VF-VaryOff, X+DRTX-CallWtrOptions,
ACTION=X-CallWtr, XR-CallWtrResched
NP    PRINTER  Status  Group  SForms  SClass  JobName  JobID
PRT002 AC        LOCAL  1PRT
PRT003 AC        LOCAL  1PRT
PRT004 OFF       LOCAL  1PRT
PRT005 OFF       LOCAL  1PRT
PRT00F OFF       LOCAL  1PRT
PRT017 OFF       LOCAL  1PRT
PRT018 OFF       LOCAL  1PRT
PRT019 OFF       LOCAL  1PRT
PRT80E OFF       LOCAL  1PRT
PRT80F OFF       LOCAL  1PRT
    
```

04/036

© 2010 IBM Corporation

This screen shot shows the JES3 printer panel with SET ACTION LONG enabled. Note that the E, S, and X parameters are abbreviated in the display. If all the combinations and permutations allowed for these had been listed, they would not fit on the screen.

IBM Systems

JES3 Printer Actions

```

SDSF PRINTER DISPLAY SY1
COMMAND INPUT ==> set action short_
SET COMMAND COMPLETE
PREFIX=* DEST=(ALL) OWNER=* SYSNAME=
ACTION=//, =, +, BC, BCn, BCnP, BD, BN, BNn, BNnP, C, CG, CJ, CP, CT, D, DL, E, EA, ED, EH, EJ, EL,
ACTION=EM, ER, ET, EX, F, Fc, FCn, FCnP, FN, FNn, FNnP, K, L, LD, S, SA, SD, SM, ST, SX, V, VF, X,
ACTION=XD, XR, XT, XX
  
```

NP	PRINTER	Status	Group	SForms	SClass	JobName	JobID
	PRT002	AC	LOCAL	1PRT		DIP	JOB00006
	PRT003	AC	LOCAL	1PRT		SDSF	JOB00007
	PRT004	OFF	LOCAL	1PRT			
	PRT005	OFF	LOCAL	1PRT			
	PRT00F	OFF	LOCAL	1PRT			
	PRT017	OFF	LOCAL	1PRT			
	PRT018	OFF	LOCAL	1PRT			
	PRT019	OFF	LOCAL	1PRT			
	PRT80E	OFF	LOCAL	1PRT			
	PRT80F	OFF	LOCAL	1PRT			
	PRT203	OFF	LOCAL	1PRT			
	PRT204	OFF	LOCAL	1PRT			
	PRT303	OFF	LOCAL	1PRT			
	PRT304	OFF	LOCAL	1PRT			
	PRT403	OFF	LOCAL	1PRT			

© 2010 IBM Corporation


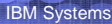
This screen shot shows the JES3 printer panel with SET ACTION SHORT enabled. Note that all of the qualifying characters on E, S, and X are listed, but not all combinations are shown. If all the combinations and permutations allowed for these had been listed, they would not fit on the screen.

The screenshot shows the SDSF PRINTER DISPLAY SY1 interface. At the top, there is a menu bar with options like 'Display', 'Filter', 'View', 'Print', 'Options', 'Search', and 'Help'. Below the menu, the command input area shows 'ACTION REQUIRED' and 'SCROLL ==> CSR' with a red circle around them. The main display area contains a table of printer configurations:

NP	PRINTER	Status	Group	SForms	SClass	JobName	JobID
	PRT002	AC	LOCAL	1PRT	ABC	DIP	JOB00006
	PRT003	AC	LOCAL	1PRT		SDSF	JOB00007
	PRT004	OFF	LOCAL	1PRT			
	PRT005	OFF	LOCAL	1PRT			
	PRT00F	OFF	LOCAL	1PRT			
	PRT017	OFF	LOCAL	1PRT			
	PRT018	OFF	LOCAL	1PRT			
	PRT019	OFF	LOCAL	1PRT			
	PRT80E	OFF	LOCAL	1PRT			
	PRT80F	OFF	LOCAL	1PRT			
	PRT203	OFF	LOCAL	1PRT			
	PRT204	OFF	LOCAL	1PRT			
	PRT303	OFF	LOCAL	1PRT			
	PRT304	OFF	LOCAL	1PRT			
	PRT403	OFF	LOCAL	1PRT			

At the bottom of the window, there is a status bar showing '10/047' and a connection message: 'Connected to remote server /host POKVMTL4.POK.IBM.COM using port 23'.

This is what you would see if you tried to overtype a field that required an action character along with the overtype. To get the overtype to work for Sclass, an action character of E or S is required. Most, but not all JES3 printer overtypes work this way.



Search Help

- Search SDSF's help panels using the SEARCH command:
 - **SEARCH phrase**
 - Up to 4 words
 - Enclose in quotes for exact match
 - For example:
 - SEARCH *message-ID*
 - SEARCH *column-title*
 - SEARCH *command-name*
 - SEARCH '*key-words*'
 - Displays a list of matches found in help
 - Select a match to display the help panel
 - ISPF, English only

© 2010 IBM Corporation

22

The screenshot shows the SDSF Search Help interface. At the top, there are menu options: Display, Filter, View, Print, Options, Search, and Help. The main display area shows a table of system statistics with columns for JOBNAME, StepName, Owner, CPU%, SPCPU%, and SLCPU%. A search for 'slcpu%' is performed, and the results are displayed in a table with columns for Title and Description. The search results show that SLCPU% represents the percentage of time the LPAR is busy for the system.

Search

```

A AOTS (ALL) PAG 0 CPU/L 19/ 19
COMMAND INPUT ==> search slcpu%
NP JOBNAME StepName Owner CPU% SPCPU% SLCPU% C
ANTMAIN ANTMAIN 0.00 19 19
    
```

Select



```

Search Help Row 1 to 3 of 3
Command ==>
Search for: SLCPU% / Show titles
/ SLCPU% Percentage of time the LPAR is busy for the system,
HELP: Display Active Users Panel -- Fields Panel 9 of 9
- in the most recent interval. The value for SLCPU%
HELP: Display Active Users Panel -- Fields Panel 9 of 9
- SLCPU%
    
```

Title	Description
SLCPU%	Percentage of time the LPAR is busy for the system, in the most recent interval. The value for SLCPU% is the same for all rows for a system. (RMF)

See

© 2010 IBM Corporation 23



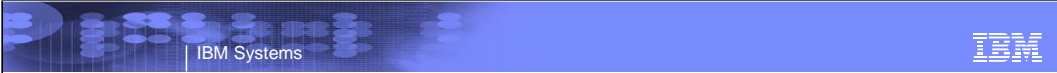
SYSLOG access via REXX

- **Problem Statement / Need Addressed:**
 - Access to SYSLOG through SDSF/REXX
 - Applies to both JES2 and JES3
- **Solution:**
 - New ISFLOG SDSF/REXX command to access syslog
- **Benefit:**
 - Simplified scanning of syslogs using SDSF/REXX execs
 - Eliminate need to allocate each syslog separately and read them
 - Allocate syslog for reading by a utility such as EXECIO
 - Read syslog records into REXX stem variables
 - Select records based on time and date

© 2010 IBM Corporation

24

SDSF/REXX was introduced in z/OS V1R9 and enhanced since. It allows access to the SDSF panels through REXX. Row and column data is represented as REXX stem variables.



ISFLOG Allocate Example

```

rc=isfcalls("on")

Address SDSF "ISFLOG ALLOCATE"
do ix=1 to isfdname.0
  "EXECIO 10 DISKR" isfdname.ix "(FINIS STEM log."
  do jx=1 to log.0
    Say mid "log."jx "is:" log.jx
  end
end
rc=isfcalls("off")

```

Annotations:

- Allocate the logical SYSLOG (points to "ISFLOG ALLOCATE")
- Loop through DD names (points to "do ix=1 to isfdname.0")
- Read contents into log. stem variable (points to "do jx=1 to log.0")
- Report the log data (points to "Say mid "log."jx "is:" log.jx")

25

© 2010 IBM Corporation

isfcalls("on") must be the first statement in all SDSF/REXX execs to define the host command environment.



As with all SDSF allocations, ISFLOG ALLOCATE returns the allocated ddname in the isfdname stem. The data set name is also available in the isfdsname stem.

This example uses EXECIO to read the allocated syslog using EXECIO. ISFLOG uses the JES logical syslog, so only a single data set will be allocated.

You can add logic to scan for specific messages or events.

Finally, use the isfcalls("off") statement to remove the host command environment.


As with all SDSF functions, you should always check the return code from a command. Any error messages are returned in the isfmsg variable and isfmsg2 stem variable,.



Results: **ISFLOG Allocate Example ...**


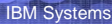
```
log.1 is: X 0000000 SY1    2010062 14:54:54.19 SYSLOG
00000000 IEE042I SYSTEM LOG DATA SET INITIALIZED
log.2 is: N 0000000 SY1    2010062 14:53:55.57
00000290 IEA630I OPERATOR *OPLOG01 NOW ACTIVE,
SYSTEM=SY1
log.3 is: NC0000000 SY1    2010062 14:53:55.60 INTERNAL
00000290 CONTROL M,UEXIT=Y IEAVN701 - INTERNALLY
ISSUED K M
log.4 is: NR0000000 SY1    2010062 14:53:56.11 INTERNAL
00000090 IEA590I WTO USER EXIT IEAVMXIT NOT FOUND
```

... more



© 2010 IBM Corporation

26



ISFLOG Read Example

```
rc=isfcalls("on")
```

Address SDSF "ISFLOG READ"

Read the logical SYSLOG into the isfline. stem

```
do ix=1 to isfline.0
```

Say mid "isfline."left(ix,5)":" isfline.ix

Report the log data

```
end
```


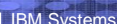
```
rc=isfcalls("off")
```

© 2010 IBM Corporation

27

This example reads the current day of syslog because the start time and date are defaulted.

Each record of syslog is loaded into the isfline stem variable. Isfline.0 has the number of variables created.



ISFLOG Special Variables


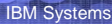
- Starting date and time
 - isflogstarttime (hh:mm:ss.th)
 - Default is 00:00:00.00
 - isflogstartdate (mm/dd/yy)
 - Default is current day
- Ending date and time
 - isflogstoptime (hh:mm:ss.th)
 - Default is 23:59:59.59
 - isflogstopdate (mm/dd/yy)
 - Default is current day
- isfdate (specify date format)

© 2010 IBM Corporation

28

You can use the time and date variables to select a portion of the log to process. You should pick reasonable limits since that affects the number of variables created.

The isfdate special variable supports the same date patterns as the interactive SET DATE command.



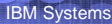

ISFLOG Special Variables ...

- isflinelim
 - Specifies the maximum number of variables to be created
 - Default is no limit
- isflinelim=10000
 - Create a maximum of 10,000 variables

© 2010 IBM Corporation

29

When the line limit is reached, SDSF stops creating variables, issues a message, and ends with a return code of 4.

ISFLOG Read Example By Time/Date

```

rc=isfcalls("on")

isfdate="mmdyyyy /"
currday=date("C")
currday=currday-1 /* yesterday */
isflogstartdate=date("U",currday,"C") /* yesterday in mm/dd/yy */
isflogstarttime=time("N") /* current time */
isflogstopdate=date("U") /* current date in mm/dd/yy */
isflogstoptime=time("N") /* current time */

isflinelim=1000

Address SDSF "ISFLOG READ"

```


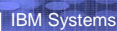
© 2010 IBM Corporation

This example reads the last 24 hours of syslog.

The `isfdate` variable is used to set the date format, in this case, `mm/dd/yyyy`. SDSF also accepts a two digit year with this format.

The current day number is obtained. We subtract 1 to get yesterday's day number and then convert it to `mm/dd/yy` format.

We set the starting and ending time to the current time.



ISFLOG Read Example By Time /Date

```
do ix=1 to isfline.0
  Say mid "isfline."left(ix,5)":" isfline.ix
end

do ix=1 to isfmsg2.0
  Say isfmsg2.ix
end

rc=isfcalls("off")
```



Report the log data

Report any messages

© 2010 IBM Corporation

31

You should always check the return code from SDSF and list the messages.



ISFLOG Read Example By Time/Date

Resulting messages:



```
ISF754I Command 'SET DATE MMDDYYYY /' generated from associated
variable ISFDATE.
ISF757I Variable ISFLINELIM being processed with value '1000'.
ISF757I Variable ISFLOGSTARTTIME being processed with value '09:34:29'.
ISF757I Variable ISFLOGSTARTDATE being processed with value '03/03/10'.
ISF757I Variable ISFLOGSTOPTIME being processed with value '09:34:29'.
ISF757I Variable ISFLOGSTOPDATE being processed with value '03/04/10'.
ISF770W Request limit 1000 from variable ISFLINELIM reached, processing
stopped.
```

© 2010 IBM Corporation

32

You should always check the return code from SDSF and list the messages.

The records from syslog will be in the isfline stem variables.


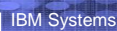


SDSF data through Java: Overview

- **Problem Statement / Need Addressed:**
 - Access SDSF through a Java application
- **Solution:**
 - Use SDSF/Java classes in your Java application
- **Benefit:**
 - Use Java for access to SDSF
 - Access panels and panel data
 - Retrieve syslog data and issue system commands
 - Retrieve job output
 - Take action to perform functions similar to action characters and overtyping
 - Filter data to reduce output returned
 - View results of interactions
 - Control access through standard SDSF security mechanisms
 - Display and modify system data

© 2010 IBM Corporation

33


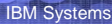


Setup

- Update CLASSPATH environment variable to reference SDSF jar file:
 - export CLASSPATH=/usr/include/java_classes/isfjcall.jar:\$CLASSPATH
- Update LIBPATH to reference SDSF DLL:
 - export LIBPATH=/usr/lib/java_runtime:\$LIBPATH (31-bit)
 - export LIBPATH=/usr/lib/java_runtime64:\$LIBPATH (64-bit)
- SDSF requires Java SDK V6
 - Either 31-bit or 64-bit mode

© 2010 IBM Corporation

34


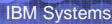


Writing a Java Application

- Create a runner that corresponds to the panel you want to work with
 - A runner is a Java class that provides access to SDSF
 - Contains a results object describing completion of request
- Create request settings and associate it with runner
- Invoke SDSF to create a list of objects
- Process the returned objects and obtain column values for each row
- Invoke methods on a row object to retrieve information or modify the object

© 2010 IBM Corporation

35



Example Java Application

```
// Create optional settings object
ISFRequestSettings settings = new ISFRequestSettings();
settings.addISFPrefix("***"); // Set job name prefix
settings.addISFOwner("ibmuser"); // Set job owner

// Get a runner used to access SDSF ST panel
ISFStatusRunner runner = new ISFStatusRunner(settings);

List<ISFStatus> statObjList = null;


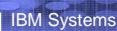
try {
    statObjList = runner.exec();
} catch (ISFException e) {
    // Process exception here
} finally {
    // Print SDSF messages related to request
    results.printMessageList(System.err);
}

// List job properties
if (statObjList != null) {
    for (ISFStatus statObj : statObjList) {
        System.out.println(statObjList.toVerboseString());
    }
}
```

© 2010 IBM Corporation

36

We will come back to this example later.


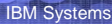


Runners and Settings

- A runner provides access to SDSF similar to SDSF commands
 - Choose the runner corresponding to the panel you want to access
 - ISFStatusRunner – ST (status panel)
 - ISFOutputRunner – O (output panel)
 - ISFHealthCheckRunner – CK (health checks)
 - etc.
 - ISFRunner – slash command, WHO, QUERY
 - Complete cross reference of runners to panels contained in the Javadoc

© 2010 IBM Corporation

37



Example Java Application

```
// Create optional settings object
ISFRequestSettings settings = new ISFRequestSettings();
settings.addISFPrefix("***"); // Set job name prefix
settings.addISFOwner("ibmuser"); // Set job owner

// Get a runner used to access SDSF ST panel
ISFStatusRunner runner = new ISFStatusRunner(settings);

List<ISFStatus> statObjList = null;

try {
    statObjList = runner.exec();
} catch (ISFException e) {
    // Process exception here
} finally {
    // Print SDSF messages related to request
    results.printMessageList(System.err);
}



// List job properties
if (statObjList != null) {
    for (ISFStatus statObj : statObjList) {
        System.out.println(statObj.toVerboseString());
    }
}
```

© 2010 IBM Corporation

38

To access the rows and columns on the ST panel, we get an instance of an `ISFStatusRunner`.

Note the constructor references the settings object that was created above it. You can also use the `setRequestSettings()` method to relate settings to a runner.



Runners and Settings ...

- Settings are used to qualify the request
 - Job name prefix, owner, destination
 - Most settings correspond to SDSF commands
 - Limit the column values retrieved
- Represented by ISFRequestSettings class
 - Create an instance of settings and associate it with runner
 - Various addISFxxxx methods to add a setting to the object

© 2010 IBM Corporation

39

You should minimize the amount of data being returned. Always try to use the settings to limit the objects being returned, such as through prefix, owner, and filter.

Limit the amount of data within each object to just those column values needed. The addISFCols setting is used to specify a list of column names for which data is needed.

IBM Systems IBM

Runners and Settings ...

```
// Create optional settings object
ISFRequestSettings settings = new ISFRequestSettings();

settings.addISFPrefix("***");
settings.addISFOwner("ibmuser");
settings.addISFCols("jname jobid");

// Get a runner used to access SDSF ST panel using settings
ISFStatusRunner runner = new ISFStatusRunner(settings);
```

Corresponds to PREFIX ** command

Corresponds to OWNER IBMUSER command


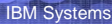
Requests just the JOBNAME and JobID columns

© 2010 IBM Corporation 40

The request settings are used to qualify the request. Most settings correspond to SDSF commands, such as PREFIX and OWNER. Use the appropriate method to add a value for the setting to the request settings object.

The object limit setting can be used to cap the number of objects returned.

Settings are associated with a runner. They remain in effect unless they are changed or reset.



Example Java Application

```
// Create optional settings object
ISFRequestSettings settings = new ISFRequestSettings();
settings.addISFPrefix("**"); // Set job name prefix
settings.addISFOwner("ibmuser"); // Set job owner

// Get a runner used to access SDSF ST panel
ISFStatusRunner runner = new ISFStatusRunner(settings);

List<ISFStatus> statObjList = null;

try {
    statObjList = runner.exec();
} catch (ISFException e) {
    // Process exception here
} finally {
    // Print SDSF messages related to request
    results.printMessageList(System.err);
}


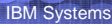
// List job properties
if (statObjList != null) {
    for (ISFStatus statObj : statObjList) {
        System.out.println(statObjList.toVerboseString());
    }
}
```

© 2010 IBM Corporation

41

Here an ISFRequestSettings object is created.

The jobname prefix is set to ** and the owner to IBMUSER. These settings correspond to the SDSF PREFIX and OWNER commands, and thus you must be authorized to change them.




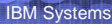
Retrieving Objects

- Panel objects are returned as lists
 - Each row represented by an object
 - Each object implements an interface for that panel
- Object interfaces
 - ISFStatus for ST row objects
 - ISFOutput for O row objects
 - ISFHealthCheck for CK objects

© 2010 IBM Corporation

42

There is an interface for each object type. Refer to the Javadoc for the complete set of object interfaces.



Retrieving Objects ...



- Use the `exec()` method to retrieve the list of objects
 - `List<ISFStatus> statObjList = runner.exec();`
- The objects returned are limited by the settings in effect
 - If `prefix=userid*`, only jobs starting with that `userid` would be returned
 - Very similar to SDSF interactive and SDSF/REXX

© 2010 IBM Corporation

43

The runner `exec()` method retrieves the objects and returns them in a `List`. Then use standard list methods to traverse the objects being returned.

If the list cannot be created, an exception will be thrown. Explanatory messages will be returned in the `ISFRequestResults` object.



Example Java Application

```
// Create optional settings object
ISFRequestSettings settings = new ISFRequestSettings();
settings.addISFPrefix("***"); // Set job name prefix
settings.addISFOwner("ibmuser"); // Set job owner

// Get a runner used to access SDSF ST panel
ISFStatusRunner runner = new ISFStatusRunner(settings);

List<ISFStatus> statObjList = null;


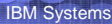
try {
    statObjList = runner.exec();
} catch (ISFException e) {
    // Process exception here
} finally {
    // Print SDSF messages related to request
    results.printMessageList(System.err);
}

// List job properties
if (statObjList != null) {
    for (ISFStatus statObj : statObjList) {
        System.out.println(statObjList.toVerboseString());
    }
}
```

© 2010 IBM Corporation

44

Notice the runner.exec() method to create the object list (of type ISFStatus).


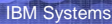


Request Results

- The runner references an ISFRequestResults object that is updated after each request
 - Contains messages describing completion of request
 - Return and reason codes
 - List of columns returned
 - Convenience methods to print messages
- Always check the results after each request
 - `ISFRequestResults results = runner.getRequestResults();`
 - `results.printMessageList();`

© 2010 IBM Corporation

45



Working with Objects


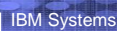
- SDSF creates one object per row
 - Column values are contained within the object
 - Use `getValue()` method to retrieve a column value
 - Use the SDSF column name (FLD name), not the column title
 - `String jobname=statObj.getValue("jname")`
 - `String owner=statObj.getValue("ownerid")`
 - Use `getFixedField` method
 - `String fixedField=statObj.getFixedField();`

© 2010 IBM Corporation

46

`getValue` always returns a formatted, string value for the column.
Other methods available are:

- `getValueBytes` – returns column value as a byte array
- `getRelatedValue` – returns an array of related values when a column contains multiple values
- `getRelatedValueBytes` – returns an array of related values as byte arrays



Working with Objects ...



```
statObjList = runner.exec();  
  
for (ISFStatus statObj : statObjList) {  
    // Get job name  
    String jobname = statObj.getJName();  
  
    // Print short form of row properties  
    System.out.println(statObj);  
  
    // Print all properties for row  
    System.out.println(statObj.toVerboseString());  
  
}
```

© 2010 IBM Corporation

47

Many panels have convenience methods to retrieve commonly used columns.

Here we see the `getJName()` method to obtain the job name and is equivalent to `getValue("jname")`.




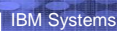
Actions

- You can modify an object similar to an action character
- Actions are represented by methods
 - Available actions defined in the interface for the object
 - Cancel a job:
 - `statObj.cancel();`
 - `results.printResponseList(System.out);`
 - The response list contains any system messages generated in response to the command
 - Similar to SDSF ULOG format

© 2010 IBM Corporation

48

The response list contains a list of system messages generated as a result of the cancel command.


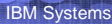


Overtypes

- You can modify an object similar to an oertype
 - Use the requestPropertyChange method
 - Method takes two input arrays:
 - Column name array
 - Column value array
 - Each column in the name array is changed to the corresponding value in the value array

© 2010 IBM Corporation

49



Overtypes ...

```
// Change job class to class A

// Build column name array
String propName = { "jclass" };

// Build column value array
String propValue = { "a" };


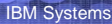
// Change the job class
statObj.requestPropertyChange(propName, propValue);

// Print response list
results.printResponseList(System.out);
```

© 2010 IBM Corporation

50

In this example, the job class for a job is changed to class A.



MVS Commands

- Can issue one or more MVS commands
- Use ISFRunner with system method
 - Takes an array of string commands
- Get responses using getRequestResults()

© 2010 IBM Corporation

51


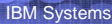
IBM Systems

Appendix

- SDSF Reference Material
 - *SDSF Operation and Customization, SA22-7670*
 - SDSF Javadoc
 - Download from `/usr/include/java_classes/isfjcallDoc.jar`
 - Unzip the jar file
 - Point your browser at `index.html`

© 2010 IBM Corporation

52

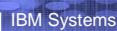



Appendix

- Information about SDSF and SDSF/REXX
 - *SDSF Operation and Customization, SA22-7670*
- SDSF REXXHELP command
 - Interactive help for SDSF/REXX
 - Many sample scripts to get you started
- SDSF SEARCH command
 - Search any SDSF help panel by phrase
- *Implementing REXX Support in SDSF Redbook, SG24-7419*

© 2010 IBM Corporation

53



Appendix – New PR Columns (JES2)

<u>New columns:</u>	
OGID1	Outgrp id 1 of current printing output
OGID2	Outgrp id 2 of current printing output
Trans	Character translation
TrkCell	De-spool the entire track cell
NewPage	Honor skip-to-channel?
SVol1-4	Spool volumes for work selection
Char1-4	Character arrangement table 1-4
HonorTRC	Whether TRC is honored from JCL
FSASysNm	MVS system where FSA is active
Line-Lim-Lo	Lower bound for selection line limit
Line-Lim-Hi	Upper bound for selection line limit
Page-Lim-Lo	Lower bound for selection page limit
Page-Lim-Hi	Upper bound for selection page limit

Items in **bold** text are overtypeable

© 2010 IBM Corporation


54



Appendix – PR Columns (JES3)

Columns:	
PRINTER	Device name (fixed field)
Group	Device group name
Status	Printer status
SForms	Forms for work selection
Sclass	Class list for work selection
Jobname	Job name of job active on printer
JobID	Job id of job active on printer
Owner	Owning userid of job active on printer
Rec-cnt	Total records (current output)
Rec-Prt	Current record
Page-Cnt	Total pages (current output)
Page-Prt	Current page
JP	Job priority of current job
DP	Output priority of current job

Items in **bold** text are overtypable

55



IBM Systems
IBM

Appendix – PR Columns (JES3)

Columns:	
SBurst	Printer selection burst mode
SPrMode1-4	Printer selection process mode 1-4
M	Mark forms control
Npro	Non-process runout time
Mode	Control mode of printer (COMP or FSS)
CkptRec	Number of records per checkpoint
CkptPage	Number of pages per checkpoint
CkptSec	Default checkpoint interval (seconds)
CkptMode	Checkpoint mode (RECS, PAGE, SEC)
CpyMod	Copy modification module ID
Unit	Printer unit
DFCB	Device default FCB
Setup	Setup mode

Items in **bold** text are overtypeable

© 2010 IBM Corporation
56





Appendix – PR Columns (JES3)

Columns:	
C	SYSOUT class of current job on printer
Seclabel	Seclabel of current job on printer
Forms	Forms of current job on printer
FCB	FCB of current job on printer
UCS	UCS of current job on printer
Flash	Flash of current job on printer
Burst	3800 burst setting of current job on printer
SepDS	Produce separator page between data sets
PrMode	Process Mode of current job on printer
SFCB	Printer selection FCB ID
SUCS	Printer selection UCS ID
SFIh	Printer selection flash ID
Work-Selection	Printer work selection criteria

Items in **bold** text are overtypeable

© 2010 IBM Corporation

57



Appendix – PR Columns (JES3)

Columns:	
CopyMark	Copymark mode
Pau	Pause between data sets
Tr	Printer tracing
FSSName	FSS defined for the printer
FSSProc	Proc used to start the FSS
SysName	System name (global)
JESN	Subsystem name
JESLevel	JES version and releas
Type	Type of job currently on printer (JOB, STC, TSU)
Trans	Data translation
Char1-4	Character arrangement table 1-4
FSASysNm	System name where FSA is active
DSPName	DSP name
DevType	Device type name

Items in **bold** text are overtypeable

© 2010 IBM Corporation

58




Appendix – PR Columns (JES3)

Columns:	
Line-Lim-Lo	Printer line limit, minimum
Line-Lim-Hi	Printer line limit, maximum
Page-Lim-Lo	Printer page limit, minimum
Page-Lim-Hi	Printer page limit, maximum
DGrpY	Process data sets destined for any local device
Dyn	Device can be started dynamically
OpLog	Log operator command actions
CGS	3800 character generation storage
B	Generate burst page
PDefault	Obtain FCB/CHARS from FSS proc
Copies	Copy count
CB	Clear printer processing indicator
TRC	Table reference character

Items in **bold** text are overtypeable

© 2010 IBM Corporation


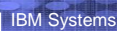
59

IBM Systems 

Appendix – PR Columns (JES3)

Columns:	
HFCB	Use designated FCB until status is changed
HChars	Use designated chars until status is changed
HUCS	Use designated UCS until status is changed
HCpyMod	Use designated CopyMod until status is changed
HFlash	Use designated Flash until status is changed
HBurst	Use designated Burst until status is changed
HForms	Use designated Forms until status is changed

© 2010 IBM Corporation 60




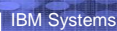
Appendix – PR Action Characters (JES3)

Action Characters:

BC	Backspace to previous checkpoint
BN	Backspace to previous internal checkpoint
BCn or BnC	Backspace number of lines from previous checkpoint
BNn or BnN	Backspace number of lines from previous internal checkpoint
BCnP	Backspace number of pages from previous checkpoint
BNnP	Backspace number of pages from previous internal checkpoint
BD	Backspace to beginning of current dataset
C	Cancel print
CG	Cancel output for current job
CJ	Cancel output of type for current job
CP	Cancel and determine page/record position of data set
CT	Cancel printer and stop
D	Display printer
DL	Display Long

© 2010 IBM Corporation

61



Appendix – PR Action Characters (JES3)


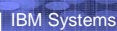
Action Characters:

E	Restart printer
EA	Restart in automatic mode
ED	Restart and turn on diagnostic mode
EH	Restart and hold
EJ	Restart and requeue all datasets for current job
EL	Restart and reload UCS, FCB, CHARS buffer
EM	Restart in manual mode
ER	Restart and perform a scheduling pass
ET	Restart and stop the printer when activity is complete
EX	Restart and turn off diagnostic mode
K	Force termination of the FSS
L	Fail the device
LD	Fail the device and take a dump

Note that multiple qualifying characters can be specified with E , for example, "EHLR". A and M are mutually exclusive. D and X are mutually exclusive.

© 2010 IBM Corporation

62






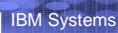
Appendix – PR Action Characters (JES3)

Action Characters:

FC	Forwardspace 0 from previous ckpt (same as BC)
FN	Forwardspace 0 from previous internal ckpt (same as BN)
FCn or FnC	Forwardspace number of lines from previous checkpoint
FNn or FnN	Forwardspace number of lines from previous internal checkpoint
FCnP	Forwardspace number of pages from previous checkpoint
FNnP	Forwardspace number of pages from previous internal checkpoint
S	Start the printer
SA	Start the printer in automatic mode
SD	Start and turn on diagnostic mode
SM	Start the printer in manual mode
ST	Start the printer and stop after current work completess
SX	Start and turn off diagnostic mode

Note that multiple qualifying characters can be specified with S , for example, "SADT". A and M are mutually exclusive. D and X are mutually exclusive






Appendix – PR Action Characters (JES3)

Action Characters:

V	Vary the printer online
VF	Vary the printer offline
X	Invoke the writer
XD	Invoke the writer and turn on diagnostic mode
XR	Invoke writer but suspend writer output until device is available
XT	Invoke writer and end after output is printerd
XX	Invoke writer and turn off diagnostic mode

Note that multiple qualifying characters can be specified with X , for example, "XDRT". D and X are mutually exclusive



64