



IBM Systems

Introduction to SDSF Support for REXX

NY Metro NaSPA Chapter meeting

November 5th, 2008

Joe Perillo, SDSF Development and Test
IBM Corporation

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.

Contents

Part 1: Brief SDSF Overview

Part 2: Introduction to SDSF Support for REXX

Part 1

SDSF Overview

System Display and Search Facility (SDSF)

Monitor, manager and control

SDSF gives you an easy and efficient way to monitor, manage and control the key aspects of your MVS/JES system. You can:

- Control job processing
- Control output, and browse jobs
- Control devices |
- Browse the syslog
- Manage system resources

```

Display  Filter  View  Print  Options  Help
-----
HGX7750 ----  SDSF PRIMARY OPTION MENU  -----
COMMAND INPUT ==>>                                SCROLL ==>> CSR

DA      Active users
I       Input queue
O       Output queue
H       Held output queue
ST      Status of jobs

LOG     System log
SR      System requests
MAS     Members in the MAS
JC      Job classes
SE      Scheduling environments
RES     WLM resources
ENC     Enclaves
PS      Processes

INIT    Initiators
PR      Printers
PUN     Punches
RDR     Readers
LINE   Lines
NODE   Nodes
SO      Spool offload
SP      Spool volumes

RM      Resource monitor
CK      Health checker

ULOG   User session log

END     Exit SDSF

```

SDSF Main menu display, with full user authority

Interactive Panels

Display Filter View Print Options Help

SDSF DA AQTs SYS1 PAG 10 SIO 113 CPU 114/007

COMMAND INPUT ==>_ SCROLL ==> PAGE

NP	JOBNAME	SysName	Real	Paging	SIO	CPU%	SrvClass
	MASTER	AQFT	10T	0.00	7.06	0.15	SYSTEM
	MASTER	AQTS	3594	0.00	0.12	0.04	SYSTEM
	ABOWEN	AQTS	742	0.00	0.00	0.00	TSOPRIME
	ADAM	AQTS	1310	0.00	0.00	0.00	TSOPRIME
	ADANPL	AQTS	1128	0.00	0.00	0.00	TSOPRIME
	ADINELL	AQTS	564	0.00	0.00	0.00	TSOPRIME
	ADOOLEY	AQFT	1472	0.00	0.00	0.00	TSOPRIME
	ADSM	AQTS	9544	0.00	0.00	0.59	SYSSTC
	AFOSTER	AQFT	1261	0.00	0.00	0.00	TSOPRIME
	AFUREY	AQTS	5496	0.00	0.00	0.00	TSOPRIME

Select a function; no learning curve

Authorized users can issue system commands from the command line

Type short commands to take actions; no need to remember command syntax

Modify values by overtyping

Control Jobs

- Immediate, up-to-date, sysplex-wide information about jobs waiting to be processed or in execution.
- Display all active jobs, all jobs on a specific queue, or display detail for a job no matter where it is in the sysplex.
- Display output as it is created

```

Display Filter View Print Options Help
-----
SDSF STATUS DISPLAY ALL CLASSES      LINE 1-20 (24651)
COMMAND INPUT ===>_                  SCROLL ===> PAGE
NP  JOBNAME  JobID   Owner   Prty Queue  C
    PEKOW2   JOB19199 SCHED   15  INPUT   J
    ACTAPSO2 JOB23991 SCHED   15  INPUT   J
    SCHEDULC JOB23474 ???????? 10  XMITTER  J
    SCHEDULC JOB31939 ???????? 10  XMITTER  J
    SCHEDULC JOB03103 ???????? 10  XMITTER  J
    IBMUSERU JOB09173 RACF2ND  7  EXECUTION A
    AGRAFA   JOB13490 AGRAF    7  EXECUTION A
    XWALLET3 JOB04538 XWALLET  7  EXECUTION B

```

Status Panel - jobs on any queue

Manage work with commands and action characters

- Cancel, hold or release jobs
- Filter the jobs to show just the jobs that interest you
- View output
- Change a job's priority, class, or destination
- Edit and resubmit the JCL without leaving SDSF

```

Display  Filter  View  Print  Options  Help
-----
SDSF INPUT QUEUE DISPLAY ALL CLASSES  LINE 22-42 (69)
COMMAND INPUT ===>                      SCROLL ===> PAGE
NP  JOBNAME  JobID   Owner    Prty C  Pos  PrtDest
   #CLRLCG  JOB26627 CLRUSER   8 D   18  LOCAL
   #CLRLCG  JOB26658 CLRUSER   8 D   19  LOCAL
   #CLRLCG  JOB26693 CLRUSER   8 D   20  LOCAL
   JYINBKUP JOB27326 ++++++++ 7 D   21  LOCAL
   JYINFPS  JOB32332 ++++++++ 7 D   22  LOCAL
   JYINFPS  JOB32333 ++++++++ 7 D   23  LOCAL
   JYINS    JOB32337 ++++++++ 7 D   24  LOCAL
  C  WLMBKP#  JOB32343 THERESA  7 D   25  LOCAL
     WLMBKP#  JOB30138 THERESA  7 G    1  LOCAL

```

Canceling a job on the Input Queue

Output

- SDSF allows you to control how and where your output is printed.
- Hold or purge the output
- Change the output's class, destination or forms

```

Display Filter View Print Options Help
-----
SDSF HELD OUTPUT DISPLAY ALL CLASSES LINES 20,599
COMMAND INPUT ===>
NP  JOBNAME  JobID  Owner  Prty C ODisp Dest
//  D16DRS1A  JOB16330 DLR      7 H HOLD P2D436
    HASCUBES  JOB15536 DLR      7 C HOLD LOCAL
    DLRTTEST  JOB16339 DLR      7 H HOLD LOCAL
    DLRRPTA   JOB17832 DLR      7 H HOLD LOCAL
    DLRRPTB   JOB17848 DLR      7 H HOLD LOCAL
//  DLREXT    JOB17888 DLR      7 H HOLD LOCAL

```

Changing the destination on multiple rows

Output

- Work with the individual output data sets for a job
- Purge or release a data set
- Browse or print a data set
- Change the class, destination or output descriptors of a data set

```
Display Filter View Print Options Help
-----
SDSF JOB DATA SET DISPLAY - JOB BERKEA5 (JOB00042)
COMMAND INPUT ===>                SCROLL ===> PAGE
NP DDNAME  StepName ProcStep DSID Owner  C Dest
   JESJCLIN                1 BEVK   R LOCAL
   JESMSGLG JES2           2 BEVK   R LOCAL
   JESJCL   JES2           3 BEVK   R LOCAL
   JESYSMSG JES2           4 BEVK   R LOCAL
   $INTTEXT JES2           5 BEVK   A LOCAL
   ISFIN    FINDDA         101 BEVK   R LOCAL
xdc ISFOUT  FINDDA         104 BEVK   R LOCAL
```

Printing an individual output dataset on the Job Data Set Panel

Browse Output

- You can display job output before deciding to print it.
- Print or search output. Or exploit the power of ISPF and use ISPF's edit or browse function to view output.
- SDSF shows the job's SYSOUT data sets the JES2 job log, the JCL for the job, and any job-related messages.

```
SDSF OUTPUT DISPLAY BKELLERZ JOB26263  DSID      2
COMMAND INPUT ===>_
      J E S 2  J O B  L O G  --  S Y S T E M  A Q F T
-----
09.25.05 JOB26263  IRR010I  USERID BKELLER  IS ASSIGNED
09.25.06 JOB26263  ICH70001I BKELLER  LAST ACCESS AT 09:
09.25.06 JOB26263  $HASP373 BKELLERZ  STARTED - INIT 12 -
09.25.07 JOB26263  IEF403I  BKELLERZ  - STARTED - TIME=09.
-----
      1 //BKELLERZ  JOB  '141691,B001D49A','WR KELLER'
        //          MSGCLASS=H,NOTIFY=BKELLER,CLASS=T
      2 //OUT OUTPUT FORMDEF=010111,PAGEDEF=V06481
        //*
        //*      THIS JOB COMPARES 2 DATA SETS.
      3 //COMPARE  PROC  NAME=,
        //          FORMAT=FULL,
-----
      16 IEF001I  PROCEDURE COMPARE WAS EXPANDED USING
ICH70001I BKELLER  LAST ACCESS AT 09:25:06 ON MONDAY,
IEF236I  ALLOC. FOR BKELLERZ COMPARE ZACTION
```

Excerpts from an Output Dataset Panel

With SDSF, you can...

- Control Devices, local and remote printers, punches, and readers
- Start and stop initiators
- Control Lines
- Manage Spool volumes, spool offloads, and a Multi-Access Spool (MAS)
- Control and Manage Nodes
- Display WLM Enclaves
- Control z/OS Unix System Services Processes
- Monitor and manage Health Checker

Part 2

SDSF and REXX

SDSF and REXX

- z/OS V1R9 SDSF adds support for the REXX programming language
 - z/OS V1R9 GA September, 2007
- Use REXX to quickly develop scripts to perform complex and repetitive tasks
- Little or no training needed if you already know SDSF
- Simpler and more powerful alternative to SDSF batch
 - Lets you include logic
- Answers or partially answers 14 customer requirements

SDSF and REXX

- With SDSF's REXX, you can perform most of the tasks that you can perform interactively, such as:
 - Display and modify jobs
 - Display and modify resources and devices
 - Browse SYSOUT data sets
 - Print SYSOUT data sets
- Use the same panel commands, action characters and column overtypes as with interactive SDSF

Getting Started

In a basic SDSF REXX exec, you:

1. Add the REXX host command environment; before issuing any SDSF commands, using **ISFCALLS**
 2. Issue an SDSF command to access a panel, using **ISFEXEC**
 3. Issue an action character or “overtyping” a column using **ISFACT**
- Data is returned in stem variables
 - Use new special variables to control results
 - These correspond to SDSF commands such as PREFIX and OWNER

Quick Start Example – Cancel a Job

```
rc=isfcalls("ON")
Address SDSF "ISFEXEC ST"
do ix=1 to JNAME.0 (variable names same as FLD names)
  if pos("KEN",JNAME.ix) = 1 then
    Address SDSF "ISFACT ST TOKEN("TOKEN.ix") PARM("NP P")
    [...lines omitted...]
rc=isfcalls("OFF")
```

Add host command environment

Access the ST panel

Find the job

Take an action on the job

Remove the host command environment (after closing the loop)

Accessing an SDSF Panel with REXX

- Use ISFEXEC to access a panel
- Syntax:
Address SDSF "ISFEXEC *sdsf-command* (*options*)"
 - *sdsf-command* is the same SDSF command as you use interactively, including parameters, for example:
 - Address SDSF "ISFEXEC DA"
 - Address SDSF "ISFEXEC CK ALL"

DA command

CK command with the ALL parameter

Accessing an SDSF Panel - Options

Options you can use when accessing a panel with ISFEXEC or ISFACT:

- PREFIX: specify a prefix for variables that are created
- PRIMARY: use the primary field list
- ALTERNATE: use the alternate field list
- DELAYED: include delayed-access columns
- NOMODIFY: don't return row tokens for use in modifying values
- VERBOSE: add diagnostic messages to the isfmsg2.stem variable (more on this later)

Accessing an SDSF Panel - Data

- SDSF builds stem variables that correspond to the panel's rows and columns
 - *column-name.index* format
 - *column-name* is the name used on an FLDENT statement (not the column title), for example:
FLDENT COLUMN(**OWNERID**),TITLE(OWNER),WIDTH(8)
 - *index* is the number of the row
 - 0 index is the number of variables in the stem
- Display the column names with the COLSHELP command

Stem Variables for Panel Data - Example

Stem variables and values for columns on the Status panel:

JNAME.0=2

JNAME.1=KENA

JNAME.2=BOBB

OWNERID.0=2

OWNERID.1=KEN

OWNERID.2=BOB

... and so on

Count of job name variables

Job name for row 1

Job name for row 2

Count of owner variables

Special Variables to Control SDSF

- Special variables for use with SDSF REXX
 - Defined by SDSF
 - Some correspond to SDSF commands
 - Others provide access to fields or data, such as the title line on an SDSF panel
 - Names start with “ISF”

Special Variables – Limit Panel Response

- Special variables with panel commands:
 - Limit the response when accessing a panel
 - Use before invoking ISFEXEC or ISFACT

- Examples

isfprefix=*

Corresponds to the command PREFIX *

isfowner=ken

Corresponds to the command OWNER KEN

isffilter=jprio gt 5

Corresponds to the command
FILTER PRTY GT 5

isfcols="JNAME JOBID OWNERID ACTSYS"

Limits the column variables created

isfsort = "TGNUM D"

Corresponds to the command
SORT TGNUM D

Special Variables - Others

- ISTLINE – contents of the panel title line
 - ISFROWS – number of rows on the panel
 - ISFDELAY – delay limit for system command responses (like SET DELAY command)
 - ISFACTIONS – controls the list of valid action characters (like SET ACTION command)
 - ISFSERVER – names the SDSF server to be used (like the server name on the SDSF command)
- ... many more

A More Complete REXX Example

```
rc=isfcalls("ON")
Address SDSF "ISFEXEC ST"
if rc<>0 then Exit 20
fixedfield=word(isfcols,1)
say "Number of rows returned" isfrows
do ix=1 to isfrows
  say "Now processing job:" value(fixedField"."ix)
  do jx=1 to words(isfcols)
    col=word(isfcols,jx)
    say " Column" col"."ix "has the value:" value(col"."ix)
  end
end
end
rc=isfcalls("OFF")
```

Variable **isfcols** contains a list of columns, with fixed field first

Variable **isfrows** contains a count of rows

A More Complete Example – Output from the Exec

Number of rows returned: 35

Now processing job: RJONES

Column JNAME.1 has the value: RJONES

Column TOKEN.1 has the value: 2hg0EwU7o5

Column JOBID.1 has the value: T0000034

Column OWNERID.1 has the value: RJONES

Column JPRIO.1 has the value: 15

Column QUEUE.1 has the value: EXECUTION

... and so on

Now processing job: TCAS

Column JNAME.2 has the value: TCAS

... and so on

Message Variables

- Message variables contain SDSF messages
 - **isfmsg** contains the SDSF short message (displayed in the upper right corner on an SDSF panel)
 - **isfmsg2.** stem contains the SDSF numbered messages
 - **Isfulog** stem is for the user log (ULOG)
- Check after each SDSF request to ensure the request was successful

Message Variables Example with Slash

Address SDSF "ISFEXEC '/\$da' (WAIT"

Issue the w/\$da command with WAIT option

if isfmsg<> "" then

Check for a short message

 Say "isfmsg is:" isfmsg

do ix=1 to isfmsg2.0

Check for a numbered message. The 0 stem contains a count of the numbered messages.

 Say "isfmsg2."ix "is:" isfmsg2.ix

end

do ix=1 to isfulog.0

Check the ULOG

 Say "isfulog."ix "is" isfulog.ix

end

Actions

Should also check the return code from the SDSF command, for example: if rc<>0 then ...

Return codes for ISFEXEC and ISFACT:

- **00** The request completed successfully.
- **08** An incorrect or invalid parameter was specified for an option or command.
- **12** A syntax error occurred parsing a host environment command.
- **16** The user is not authorized to invoke SDSF.
- **20** A request failed due to an environmental error.
- **24** A request failed due to an environmental error.

Actions

- Use the ISFACT command to issue an action character or modify a value (overtyping a column)

- Syntax:

Address SDSF "ISFACT *sdsf-cmd* TOKEN("*token*")
PARM(*parms*) (*options*)"

- *sdsf-cmd* is the same SDSF command you used with ISFEXEC to access the panel
- *token* identifies the row
- *parms* identifies the column and action

Actions - continued

TOKEN(*token*)

- TOKEN stem is returned by ISFEXEC
- Each row is identified with a TOKEN variable
- TOKEN is used as input to ISFACT

PARM(*parms*)

- Describes the action or modification
- *column value* format, for example:
 - PARM(OCLASS A FORMS 1234)
 - PARM(NP C)

Change both
class & forms

Use NP for action characters

Example - Change Output Forms

```
isfprefix="**"
```

Set filters

```
isfowner="RJONES"
```

Access O panel to set variables

```
Address SDSF "ISFEXEC O"
```

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

Find a row with job name BOB

```
if pos("BOB",JNAME.ix) = 1 then
```

```
do
```

```
Address SDSF "ISFACT O TOKEN("TOKEN.ix")  
PARM(FORMS 1234)"
```

Use the token for that row to identify it, enclosing it in single quotes

```
end
```

Change forms

```
end
```


Browse Job Data Sets

- Use ISFACT to issue the SA action character against a job
 - Allocates the data set (free=close)
 - REXX only
- Allocated ddname is returned in **isfddname.** stem variable
- Data set name is in **isfdsname.** stem variable
- Use EXECIO to read the data set

Example: Browse Job Data Sets

Address SDSF "ISFEXEC ST"

Access the ST panel, then use logic to find a job (not shown)

...

Address SDSF "ISFACT ST TOKEN("TOKEN.ix") PARM(NP SA)"

Issue SA action

do jx=1 to isfddname.0

Loop through ddnames

Say "Now reading" isfdsname.jx

"EXECIO * DISKR" isfddname.jx "(STEM line. FINIS"

Say "Lines read" line.0

EXECIO reads the data set

do kx=1 to line.0

Say " line."kx "is:" line.kx

end

end

Browse Health Check Output

- Use ISFEXEC to access the CK panel
- Use ISFACT to issue the S action character to browse check output
 - Output is placed in the **isfline.** stem special variable

Printing a Job Data Set

- Print a job data set
 - Use X, XS, XD action characters
 - ISFACT PARM(NP X)
- Attributes of print data set are controlled through special variables, for example:
 - isfpqrtclass="U"
 - isfpqrtcopies="2"
 - isfpqrtdest="ken"
 - isfpqrtformdef="fff"
 - isfpqrtforms="8888"
 - isfpqrtpagedef="pppp"
 - isfpqrtprmode="pmode"

Avoiding Duplicate Variable Names

Use the **PREFIX** option on ISFEXEC and ISFACT to add a prefix to variable names created by SDSF

- Prevents duplicate variable names in existing scripts
 - Needed when accessing the job data set panel, so that column variables don't conflict
- Format: (PREFIX *prefix*)

Example: Using the PREFIX Option

```
Address SDSF "ISFACT ST TOKEN("TOKEN.ix") PARM(NP '?')  
  (PREFIX jds_)"
```

Access JDS using NP ? and define a prefix for all JDS variables.

```
do jx=1 to jds_DDNAME.0  
  say "DSName is" jds_DSNAME.jx  
  Say "Stepname is" jds_STEPN.jx  
  Say "Procstep is" jds_PROCS.jx  
end
```

References to variables all include the prefix

Using SDSF with SYSREXX

- SDSF REXX Support works with System REXX
- Need proper security environment to access SDSF
 - Logon from console to get security environment
 - Need access to all commands used by EXEC
- Need to specify ISFJESNAME or ISFSERVER
 - ISFSERVER defaults to 'SDSF'

JES3 Support (z/OS 1.10)

- Different subset of panels/columns are available
- Use output from WHO command to determine JES being processed

```
rc=isfcalls("on")
Address SDSF "ISFEXEC WHO"
do ix=1 to isfresp.0
  if pos("JESTYPE=",isfresp.ix) <> 0 then
    do
      Parse var isfresp.ix kw "=" jestype
      Say jestype
    end
  end
end
if (jestype = "JES3") then ...
else ...
rc=isfcalls("off")
```


Security

- SDSF security applies to using SDSF through REXX
- No changes to ISFPARMS or SAF
- IBM recommends SAF for security instead of ISFPARMS for better control and auditing

