IBM Education Assistance for z/OS V2R1

Item: SDSF Enhancements
Element/Component: SDSF

Material is current as of June 2013

Agenda

- Trademarks
- Presentation Objectives
- Overview
- Usage & Invocation
- Migration & Coexistence Considerations
- Presentation Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.

Presentation Objectives

- New SDSF support for release V2R1:
  - 64-bit support
  - 8-character job class
  - Print enhancements
  - Sort enhancements
  - JESPlex scoping
  - System symbolics support
  - REXX/Java enhancements (BROWSE and LOG)
  - Security Assist

Overview: 64-bit Support

- **Problem Statement / Need Addressed**
  - Numbers of rows for many displays continues to increase as the number of jobs and output elements allowed increases
  - SDSF receives ABEND878 and other symptoms due to tables getting too large, or simply due to storage fragmentation
- **Solution**
  - SDSF row data moved to 64-bit storage
  - 64-bit storage requested on SSI 80 (job information)
  - ULOG buffers moved to 64-bit cell pool
- **Benefit / Value**
  - Larger tables supported; storage fragmentation issues mitigated.

ISF121I Message: 64-bit Support

- New ISF121I dialog message added when unable to obtain 64-bit storage
  - Issued once per session, after first failure
  - Will attempt to obtain 31-bit storage instead
    - If storage is not available request will result in ABEND if storage request was unconditional

ISF121I Module *modname* was unable to obtain *nnnnnnnn_nnnnnnnn* bytes of storage (*nnn* segments). Check MEMLIM value.

Overview: 8-character Job Class

- **Problem Statement / Need Addressed**
  - One character for job class is inadequate
  - JES2 supports 8-character classes starting in z/OS V2R1
    - JES3 has always supported them
- **Solution**
  - Displays updated to support 8-character job class in JES2 environment
- **Benefit / Value**
  - Longer job class names supported in both JES2 and JES3 environment

JC panel: 8-character Job Class

- Column Changes
  - **ACTIVE**
    - Indicates whether job class is active (YES/NO)
    - Active status also controls highlighting of row
    - Overtypeable ($TJOBCLASS,ACTIVE=)
  - **GROUP**
    - Job class group
    - Now applies to both JES2 and JES3
    - Overtypeable ($TJOBCLASS,GROUP=)
    - By default is not at end of line but after STATUS column (Same location for both JES2 and JES3 display)
- Action changes
  - **ST** action is not allowed for multi-character job classes other than STC and TSU due to ST command syntax restrictions

New and changed columns on other panels: 8-character Job Class

- **DA**, **I**, **ST**, **O**, and **H** panels
  - The job class column (**JCLASS**) is now 8 characters under JES2
- **INIT** panel
  - Existing **ICLASS** column represents list of classes
  - New **ICLASS1-8** columns display first 8 classes
  - ICLASS is overtypeable
    - Overtype extension (**+**) allows up to 36 classes
  - ICLASS1 is overtypeable
    - Overtype extension (**+**) allows up to **8** classes (or class groups)
  - When multi-character job classes or groups are present in class list:
    - ICLASS column is no longer overtypeable for that initiator
    - Any multi-character classes display as a period in ICLASS list
- **SO** panel
  - **SCLASS**, **SCLASS1-8** behave like ICLASS, ICLASS1-8 on INIT

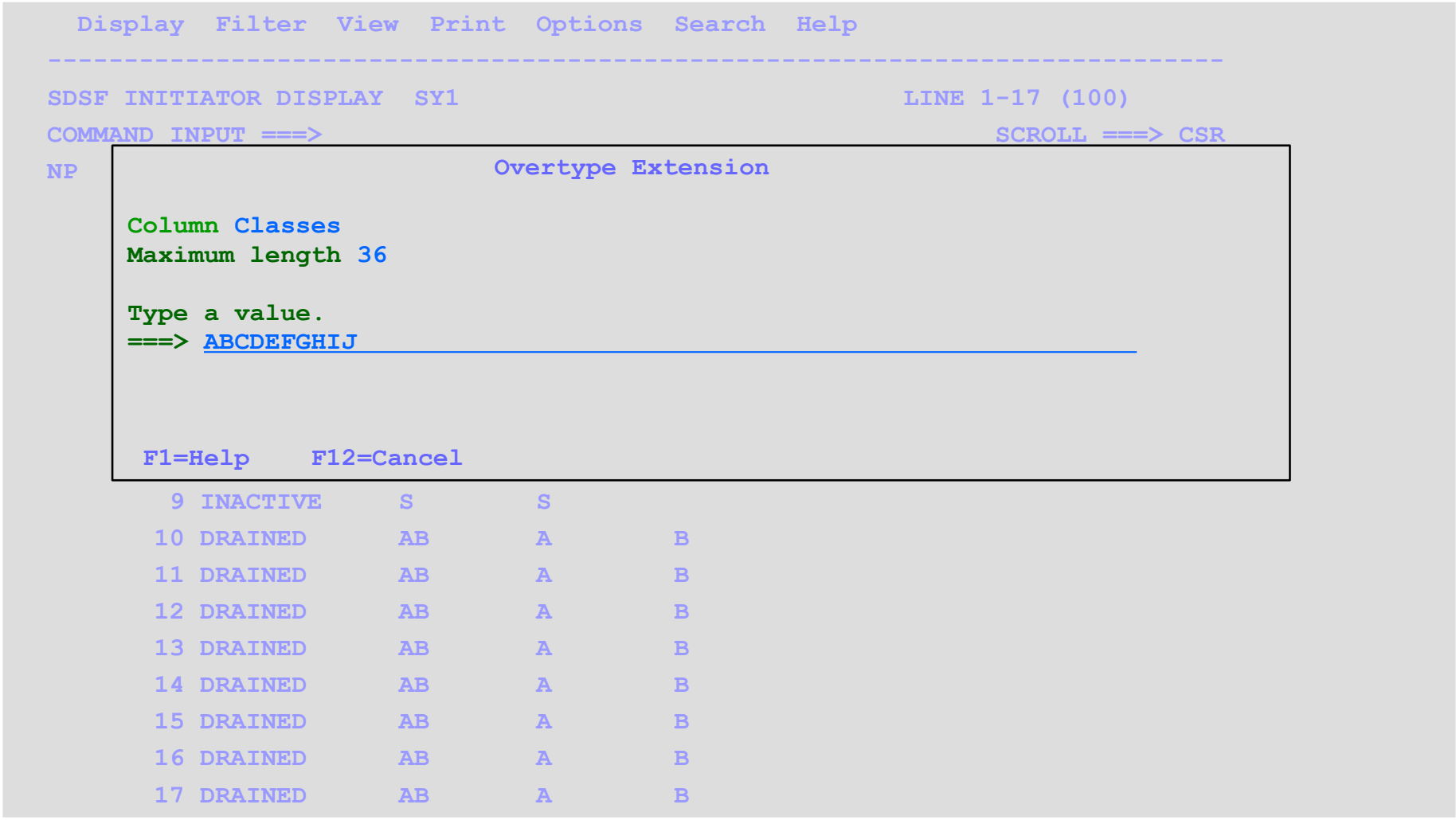
INIT panel example: 8-character Job Class

```
  Display  Filter  View  Print  Options  Search  Help
-------------------------------------------------------------------------------
SDSF INITIATOR DISPLAY  SY1                          LINE 1-17 (100)
COMMAND INPUT ===>                                     SCROLL ===> CSR
NP      ID Status       Classes  Class1   Class2   Class3   Class4   Class5
         1 INACTIVE     ..ABC    FRED     WILMA    A        B        C
         2 INACTIVE     N        N
         3 DRAINED      A        A
         4 DRAINED      JIFAB74  J        I        F        A        B
         5 DRAINED      EB74A    E        B        7        4        A
         6 DRAINED      BAG      B        A        G
         7 DRAINED      AB       A        B
         8 DRAINED      GAB      G        A        B
         9 INACTIVE     S        S
        10 DRAINED      AB       A        B
        11 DRAINED      AB       A        B
        12 DRAINED      AB       A        B
        13 DRAINED      AB       A        B
        14 DRAINED      AB       A        B
        15 DRAINED      AB       A        B
        16 DRAINED      AB       A        B
        17 DRAINED      AB       A        B
```

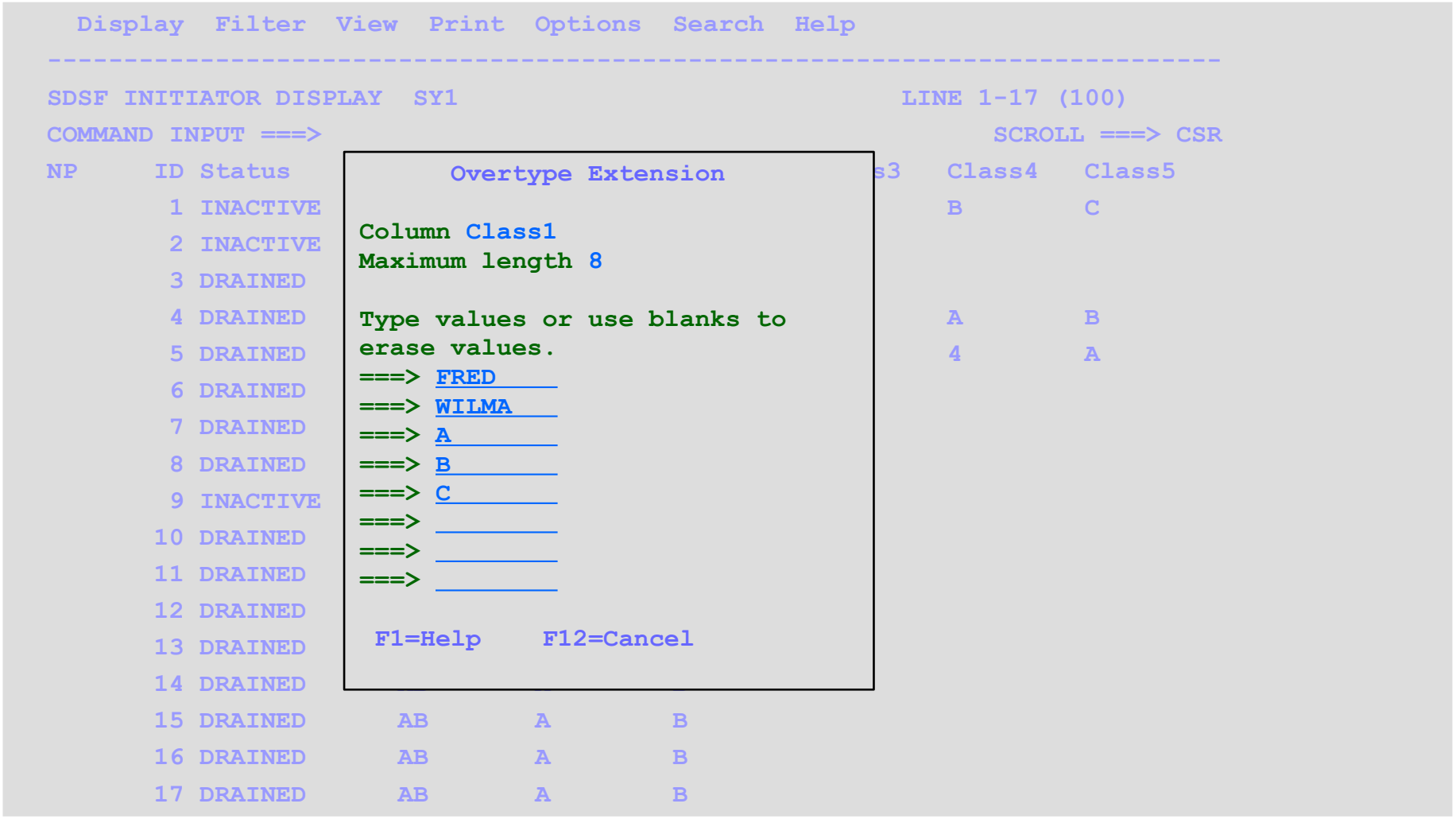
Class list overtype not allowed

FRED and WILMA replaced by .

INIT panel overtype extension example: 8-character Job Class



INIT panel overtype extension example: 8-character Job Class



Overview: Print Enhancements

- **Problem Statement / Need Addressed**
  - Printing to SYSOUT always allocates RECVM=VBA, LRECL=240
  - Printing always uses ASA carriage control
  - No ability to specify writer name when printing to SYSOUT
  - Print “as is” functionality is desirable
- **Solution**
  - Print to SYSOUT now allows RECFM, LRECL, and writer name
  - Print to SYSOUT can use source data SYSOUT attributes
  - Carriage control based on RECFM of destination data set
- **Benefit / Value**
  - Print “as is” is now possible

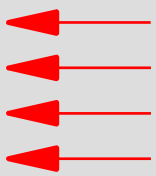
Print to SYSOUT Panel: Print Enhancements

```
                                SDSF Open Print
COMMAND INPUT ===>                                          SCROLL ===> CSR


Enter SYSOUT attributes below:

Class          ===>          (A through Z, 0 through 9)
Copies         ===>          (1 to 255)
Forms          ===>          **
Destination    ===>
FCB            ===>
UCS            ===>
Process Mode   ===>          **
Pagedef        ===>          **
Formdef        ===>          **

Output Descriptor Name ===>          (Omit with fields marked with **)
Writer name            ===>  NEW     **
Record format          ===>  NEW
Record length          ===>  NEW
Use source attributes  ===>  NEW     (YES or NO)
```



Print to SYSOUT panel: Print Enhancements

- **Writer name**
  - Specifies the writer name to be associated with the SYSOUT
- **Record format**
  - Specifies the RECFM of the SYSOUT (Default: VBA)
- **Record length**
  - Specifies the LRECL of the SYSOUT (Default: 240)
- **Use source attributes**
  - Attempts to obtain the following attributes from the source data:
    - Class, forms, destination, FCB, UCS, writer, process mode, RECFM, and LRECL
  - Panel is redisplayed with obtained values
  - Some properties may be unavailable in some contexts
  - LRECL is maximum LRECL of all datasets selected

Rexx/Java: Print Enhancements

- New Rexx variables
  - **isfprtwritername** – specifies the writer name for a print to sysout request
  - **isfprtsourceatts** = 'YES' or 'NO' – indicates whether source attributes are to be fetched on a print to SYSOUT
    - Explicitly specified attributes will be honored even if isfsourceatts='YES'
  - **isfprtrecfm** and **isfprtlrecl** – now honored for print to SYSOUT
- New Java methods
  - **addISFPrtWriter()**, **removeISFPrtWriter()**
  - **addISFPrtSourceAttributes()**, **removeISFPrtSourceAttributes()**

Carriage control: Print Enhancements

- Prior to z/OS V2R1, all printed data received ASA carriage control
  - Machine carriage control was converted to ASA
  - ASA carriage control was inserted if none existed
- In z/OS V2R1, new function is available for printed data to receive carriage control based on the RECFM of output data set (Print SYSOUT and Print Data Set)
  - RECFM ending in A (VBA, FA, etc.) receives ASA carriage control
    - Converted from machine or inserted if necessary
  - RECFM ending in M (VBM, FM, etc.) receives machine carriage control
    - Converted from ASA or inserted if necessary
  - All others (VB, F, etc) receive no carriage control
    - Machine or ASA carriage control is stripped
  - New custom property **Print.CCTL.AlwaysUseASA** currently defaults to TRUE but can be set to FALSE to enable the new (correct) behavior

Carriage control: Print Enhancements

- The **SET PRTCCASA** command provides a user-level ability to set the carriage control behavior, overriding the custom property described on the prior slide
- The syntax is as follows:

SET PRTCCASA ON | OFF | ?

where

ON indicates that ASA carriage control is used

OFF indicates that the RECFM setting for the output dataset be honored for carriage control

? indicates the current value is to be queried

The value of SET PRTCCASA is saved in the ISPF profile across SDSF sessions.

Overview: Extended Console Name

- **Problem Statement / Need Addressed**
  - ULOG display allocates extended console for ULOG based on either TSO userid or SET CONSOLE value
  - Multiple instances of SDSF (split screen or multiple logons) send messages to initial session's ULOG
  - SET CONSOLE can be changed manually for each but this is an inadequate solution if needed frequently
  - This can also be an issue for REXX if isfulog. is used
- **Solution**
  - If console is in use, SDSF will optionally attempt to use a different extended console name (by appending one character)
- **Benefit / Value**
  - Unique ULOGs for each session for split screen or multiple logons

SET CONMOD: Extended Console Name

- The new **SET CONMOD** command is used to control modification of the console name when the console name is in use.
- The syntax is as follows:

  SET CONMOD ON | OFF | ?

  where

  ON indicates that console name modification is to be done if activation fails due to the console name being in use

  OFF indicates that the console name is not to be modified

  ? indicates the current value is to be queried

- The value of SET CONMOD is saved in the ISPF profile across SDSF sessions.

Rexx/Java: Extended Console Name

- New Rexx variables
  - **isfconmod** – corresponds to the SET CONMOD command
    - Valid values are YES and NO
- New Java methods
  - **addISFConMod(), removeISFConmod()**

CONMOD custom properties: Extended Console Name

- **Console.EMCS.NoConMod**
  - When set to TRUE, **SET CONMOD ON** is not allowed. The first instance of the extended console will always be used.
  - Default: FALSE

- **Console.EMCS.ConModChars**
  - Specifies a list of up to 32 characters to choose from (in order) when appending a character to the extended console name
  - Valid characters are A-Z, 0-9, $, #, @
  - Default: $#@12345

Split screen example: Extended Console Name

```
  Display  Filter  View  Print  Options  Search  Help
 ------------------------------------------------------------------------------
 SDSF ULOG  CONSOLE D96CLW1                          LINE 0       COLUMNS 02- 133
 COMMAND INPUT ===> set conmod off                    _ SCROLL ===> CSR
******************************** TOP OF DATA ***********************************
SY1       2013023  18:27:20.11             ISF031I CONSOLE D96CLW1 ACTIVATED
******************************** BOTTOM OF DATA ********************************
```

Messages from both sessions go to top session's ULOG

```
  Display  Filter  View  Print  Options  Search  Help
 ------------------------------------------------------------------------------
 SDSF ULOG  CONSOLE D96CLW1                          LINE   CONS ACT ERR - IN USE
 COMMAND INPUT ===>                                          SCROLL ===> CSR
******************************** TOP OF DATA ***********************************
SY1       2013023  18:30:58.66             ISF032I CONSOLE D96CLW1$ ACTIVATE FAILED,
SY1       2013023  18:30:58.66             ISF042I CONSOLE D96CLW1 IS IN USE
******************************** BOTTOM OF DATA ********************************
```

Split screen example: Extended Console Name

```
  Display  Filter  View  Print  Options  Search  Help
 ------------------------------------------------------------------------------
 SDSF ULOG  CONSOLE D96CLW1                        LINE 0       COLUMNS 02- 133
 COMMAND INPUT ===> set conmod on                  SCROLL ===> CSR
********************************* TOP OF DATA ***********************************
SY1       2013023  18:42:33.52            ISF031I CONSOLE D96CLW1 ACTIVATED
******************************** BOTTOM OF DATA *********************************
```

Messages from top session go to top session's ULOG

```
  Display  Filter  View  Print  Options  Search  Help
 ------------------------------------------------------------------------------
 SDSF ULOG  CONSOLE D96CLW1$                       LINE 0       COLUMNS 02- 133
 COMMAND INPUT ===>                                SCROLL ===> CSR
********************************* TOP OF DATA ***********************************
SY1       2013023  18:45:28.70            ISF041I CONSOLE NAME D96CLW1 MODIFIED
SY1       2013023  18:45:28.70            ISF031I CONSOLE D96CLW1$ ACTIVATED
******************************** BOTTOM OF DATA *********************************
```

Messages from bottom session go to bottom session's ULOG

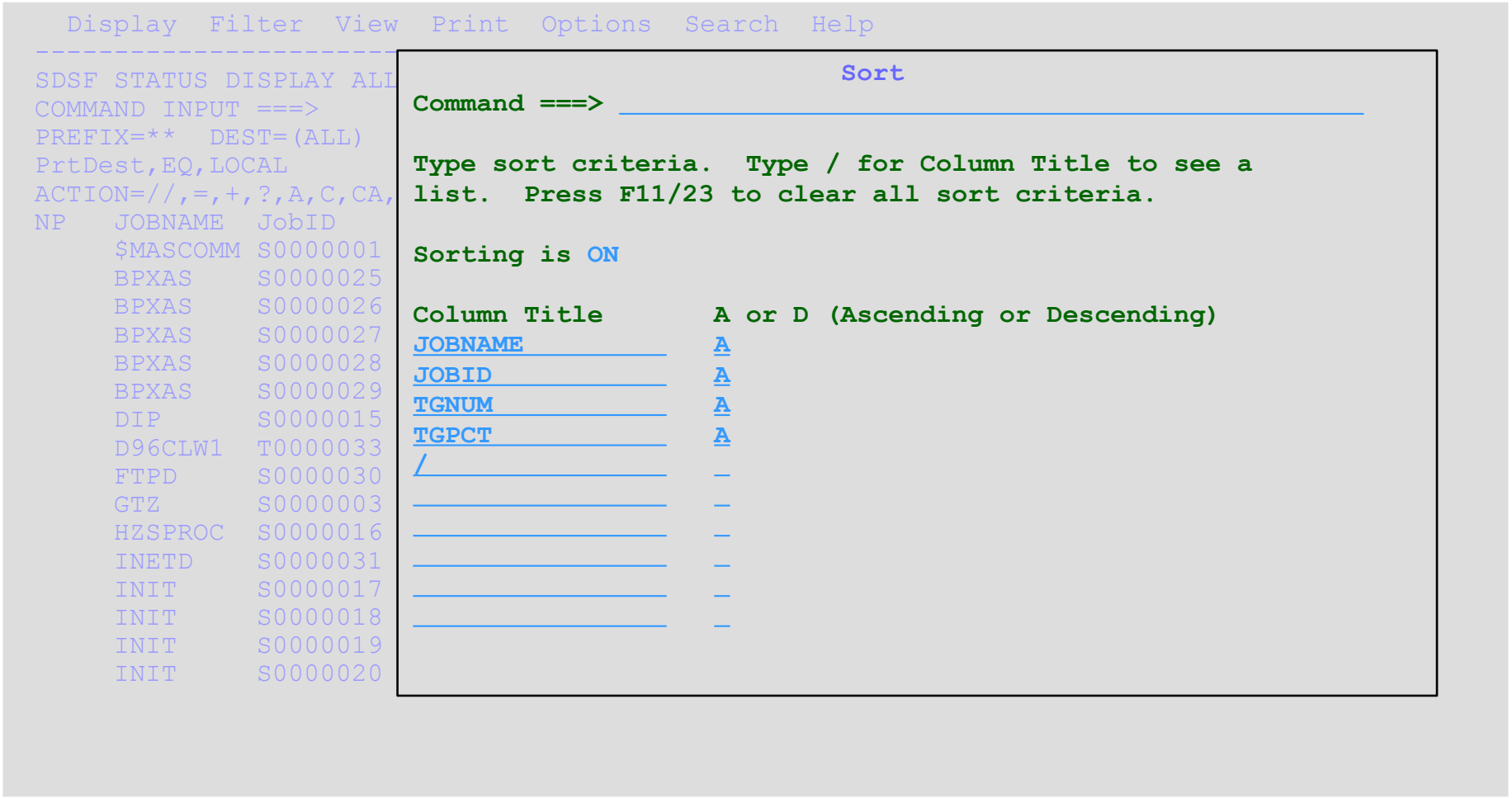
Overview: Sort Enhancements

- **Problem Statement / Need Addressed**
  - SDSF supports only 2 sort criteria at a time
  - There are scenarios where more are needed
- **Solution**
  - Number of sort criteria increased to 10
  - Additional SET DISPLAY options to display sort criteria and filters
- **Benefit / Value**
  - More flexibility in sorting panel data

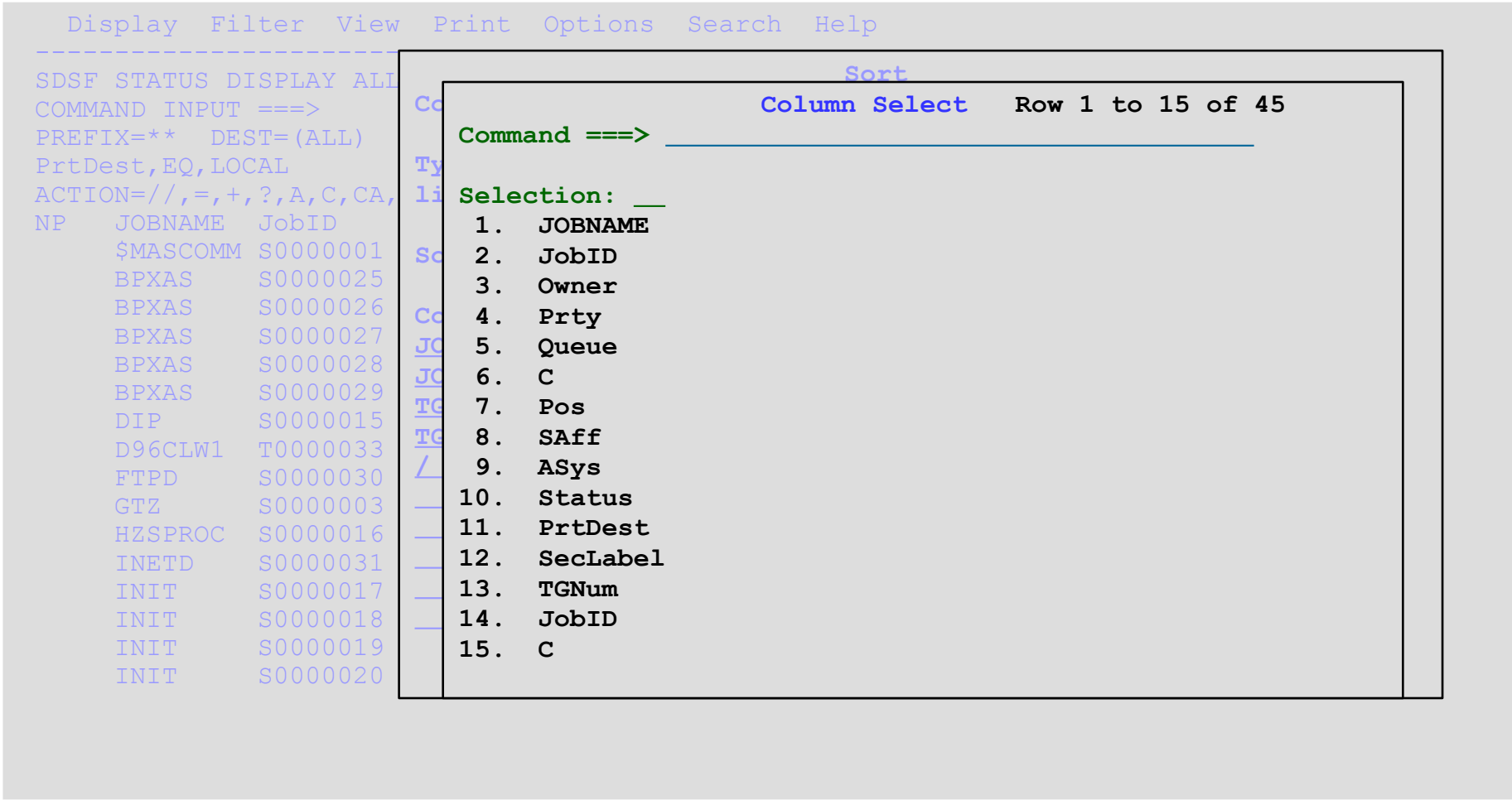
SORT Command: Sort Enhancements

- Up to 10 total sort criteria can be associated with each panel
- SORT command continues to support only two criteria at a time on command line
  - isfsort Rexx variable supports up to 10 (split into several commands)
- Sort criteria can now be qualified with a plus (+) or minus (-) to add or remove criteria from the list
  - For example
    - **SORT JOBNAME A JOBID A**
      - Sets jobname and job id as ascending sort criteria
    - **SORT +TGNUM A TGPCT A**
      - Adds TGNUM and TGPCT as a third and fourth sort criteria
- **SORT OFF** remembers prior sort criteria; **SORT ON** restores them
- Sort popup can be used to specify complete list
  - List replaces “Major” and “Minor” column specification

SORT Popup: Sort Enhancements



SORT Popup (criteria selection): Sort Enhancements



Release compatibility with >2 sort criteria: Sort Enhancements

- Prior to z/OS 2.1, all sort criteria are saved in ISPF profile as one big ISPF variable (ISFSRTC for JES2, ISFSRTC3 for JES3)
- In z/OS 2.1, sort criteria for each display are stored in separate variables (ISF2Snnn for JES2, ISF3Snnn for JES2)
  - ISFSRTC and ISFSRTC3 are still maintained for compatibility, but only populated with the first 2 sort criteria
  - If both new and compatibility variables exist,
    - The new variable is used if the first two criteria match the compatibility variable.
    - The compatibility variable is used if they are different.
  - This ensures that:
    - if the sort criteria are changed on a pre-2.1 release, the changes are honored
    - if the sort criteria are not changed on a pre-2.1 release, up to 10 sort criteria are remembered when returning to 2.1

SET DISPLAY: Sort Enhancements

- Pre-z/OS V2R1
  - **SET DISPLAY ON** shows up to 2 sort criteria and count of filters

- z/OS V2R1
  - **SET DISPLAY ON** still shows up to 2 sort criteria and count of filters
    - If more than 2 sort criteria, the number of additional criteria are listed as “+n”
    - For example, 3 criteria will display the first 2 and a “+1”
  - New **SET DISPLAY LONG** will show all sort criteria and will list all filters as “colname,op,value” triplets
    - May exceed 2 lines of display

SET DISPLAY ON example: Sort Enhancements

```
  Display  Filter  View  Print  Options  Search  Help
-------------------------------------------------------------------------------
SDSF STATUS DISPLAY ALL CLASSES                          LINE 1-15 (33)
COMMAND INPUT ===> set display on                          SCROLL ===> CSR
PREFIX=**  DEST=(ALL)  OWNER=*  SORT=JOBNAME/A JobID/A +2  SYSNAME=*
FILTERS=5
NP   JOBNAME  JobID    Owner    Prty Queue      C        Pos  SAff  ASys Statu
     $MASCOMM S0000001 JES2       15 PRINT                1
     BPXAS    S0000025 OMVSKERN    1 PRINT               12
     BPXAS    S0000026 OMVSKERN   15 EXECUTION                 SY1   SY1
     BPXAS    S0000027 OMVSKERN    1 PRINT               10
     BPXAS    S0000028 OMVSKERN   15 EXECUTION                 SY1   SY1
     B        S0000029 OMVSKERN    1 PRINT
     D        S0000030 SYSTASK     1 PRINT
     D        S0000031 SYSTASK    15 EXECUTION
     FTPD     S0000030 SYSTASK     1 PRINT                   9
     GTZ      S0000003 SYSTASK    15 EXECUTION                 SY1   SY1
     HZSPROC  S0000016 SYSTASK    15 EXECUTION                 SY1   SY1
     INETD    S0000031 SYSTASK     1 PRINT                9
     INIT     S0000017 SYSTASK    15 EXECUTION                 SY1   SY1
     INIT     S0000018 SYSTASK    15 EXECUTION                 SY1   SY1
     INIT     S0000019 SYSTASK    15 EXECUTION                 SY1   SY1
     INIT     S0000020 SYSTASK    15 EXECUTION                 SY1   SY1
     INIT     S0000021 SYSTASK    15 EXECUTION                 SY1   SY1
```

Indicates five total filters

Indicates two additional sort criteria (total of four)

SET DISPLAY LONG example: Sort Enhancements

```
  Display  Filter  View  Print  Options  Search  Help
-------------------------------------------------------------------------------
SDSF STATUS DISPLAY ALL CLASSES                          LINE 1-14 (33)
COMMAND INPUT ===> set display long                        SCROLL ===> CSR
PREFIX=**  DEST=(ALL)  OWNER=*  SORT=JOBNAME/A JobID/A TGNum/A TGPct/A
SYSNAME=*  FILTERS=TGPct,GE,0 TGNum,GE,1 JobID,EQ,* JOBNAME,EQ,*
PrtDest,EQ,LOCAL
NP   JOBNAME  JobID    Owner    Prty Queue      C         Pos  SAff  ASys Statu
     $MASCOMM S0000001 JES2       15 PRINT                 1
     BPXAS    S0000025 OMVSKERN    1 PRINT                12
     BPXAS    S0000026 OMVSKERN   15 EXECUTION                SY1   SY1
     BPXAS    S0000027 OMVSKERN    1 PRINT                10
     BPXAS                                                  ION
     BPXAS
     DIP                                                     4
     D96CLW1  T0000034 D96CLW1    15 EXECUTION              SY1   SY1
     FTPD     S0000030 SYSTASK     1 PRINT                 8
     GTZ      S0000003 SYSTASK    15 EXECUTION              SY1   SY1
     HZSPROC  S0000016 SYSTASK    15 EXECUTION              SY1   SY1
     INETD    S0000031 SYSTASK     1 PRINT                 9
     INIT     S0000017 SYSTASK    15 EXECUTION              SY1   SY1
     INIT     S0000018 SYSTASK    15 EXECUTION              SY1   SY1
     INIT     S0000019 SYSTASK    15 EXECUTION              SY1   SY1
     INIT     S0000020 SYSTASK    15 EXECUTION              SY1   SY1
```

All five filters are listed

All four sort criteria are listed

Overview: JESPlex Scoping

- **Problem Statement / Need Addressed**
  - Some SDSF panels (DA, CK, PS, ENC) have SYSPLEX scope
  - Some installations have multiple JESPlexes in a single SYSPLEX, with separate operations for each
- **Solution**
  - Panels with SYSPLEX scope can now optionally be given JESPlex scope (via custom properties)
- **Benefit / Value**
  - JESPlexes within a SYSPLEX can now be managed more independently

JESPlex Scoping

- New custom properties added
  - **Panel.DA.JESPlexScope**
  - **Panel.CK.JESPlexScope**
  - **Panel.PS.JESPlexScope**
  - **Panel.ENC.JESPlexScope**
  - **Panel.All.JESPlexScope**
    - Sets value for DA, CK, PS, ENC, and any future panels which default to SYSPLEX scope

- When set to TRUE, the corresponding panel has JESPlex scope
  - All systems on which an active member of the JESPlex exists, regardless of whether it's primary or secondary.
  - Jobs still display if running on one of these systems, even if running poly-JES and the job is not associated with the JESPlex.

Overview: Symbolics and Filtering

- **Problem Statement / Need Addressed**
  - For some display filters, it makes sense for filters to have different values on different systems (e.g. system name)
- **Solution**
  - System symbolics may now be specified within filter values
- **Benefit / Value**
  - Filters can now have different values on different systems

Using symbolics in filters: Symbolics and Filtering

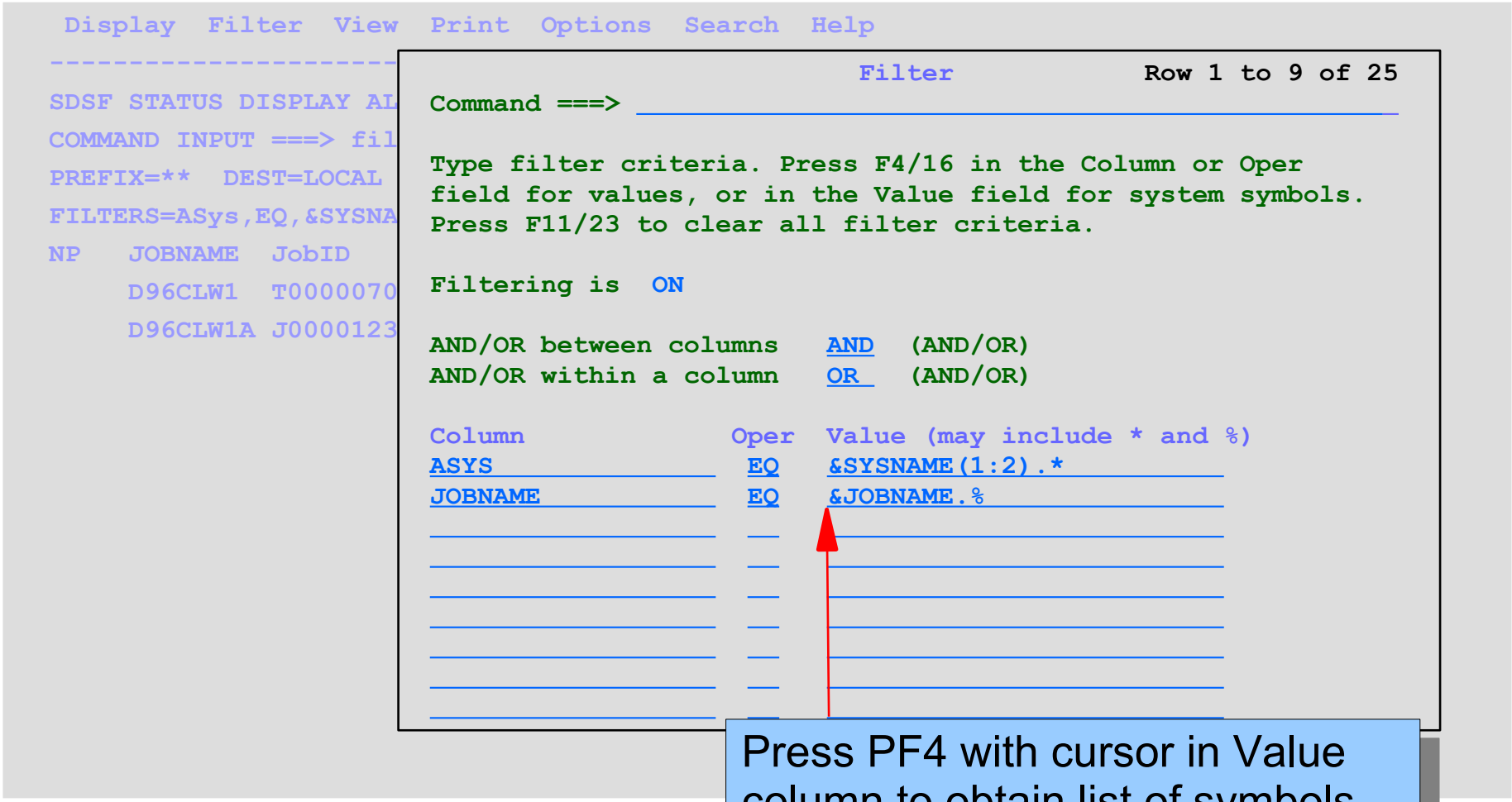
- System symbolics can now be used as the filter value, or as part of the filter value, on the SET FILTER command
- Substrings of system variables can be used (similar to JCL)
- Symbols are not substituted until the filter is applied
- For example
  - **FILTER ASYS EQ &SYSNAME**
    - Filter on all rows where ASYS is equal to the value of the local system
  - **FILTER ASYS EQ &SYSNAME(1:2).***
    - Filter on all rows where ASYS begins with the first two characters of &SYSNAME
- Symbolics available may be viewed and selected from the filter popup by selecting Prompt (PF4) with a / in the value column.
  - Current symbolic values are also shown for reference

Symbolic filter example: Symbolics and Filtering

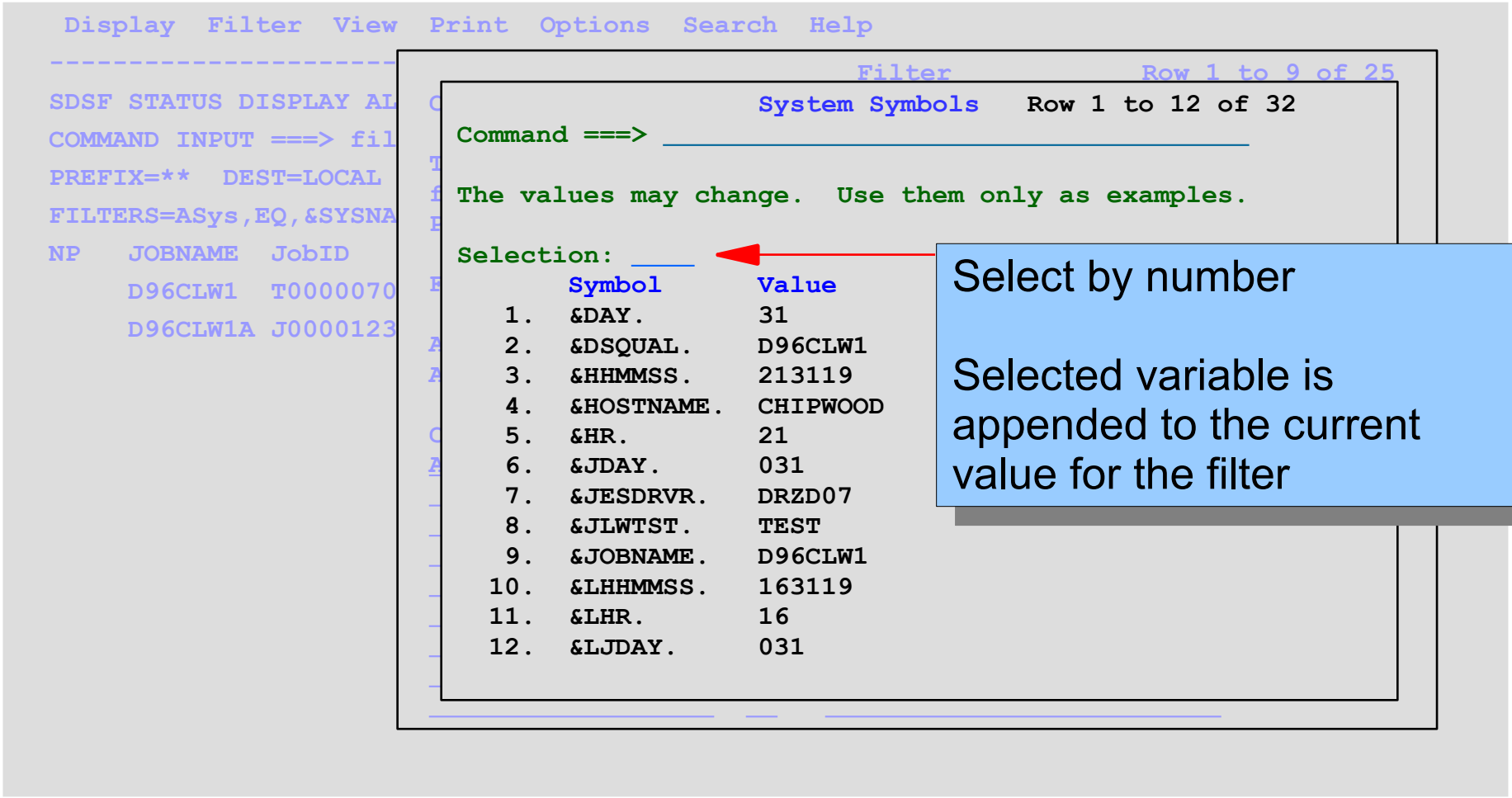
```
 Display  Filter  View  Print  Options  Search  Help
-------------------------------------------------------------------------------
SDSF STATUS DISPLAY ALL CLASSES                          LINE 1-16 (22)
COMMAND INPUT ===> filter ?                                 SCROLL ===> CSR
PREFIX=**  DEST=LOCAL  OWNER=*  SORT=Pos/A  SYSNAME=
FILTERS=ASys,EQ,&SYSNAME(1:2).* JOBNAME,EQ,&JOBNAME.%
NP   JOBNAME  JobID    Owner    Prty Queue      C          Pos  SAff  ASys Statu
     D96CLW1  T0000070 D96CLW1    15 EXECUTION                 SY1   SY1
     D96CLW1A J0000123 D96CLW1    15 EXECUTION                 SY2   SY2
```

Symbolic is not resolved until applied

Filter Popup: Symbolics and Filtering



Filter Popup: Symbolics and Filtering



System symbolic considerations: Symbolics and Filtering

- Characters **( ) :** and **.** (parens, colon, and period) are allowed as values on SET SCHARS (generic and placeholder characters)
  - If you use these characters in SCHARs, strings may change and you may not achieve the desired filter
- Symbols may be defined on one system but not another
  - If symbol is used but not defined, no substitution will occur, which may also prevent the desired filtering
- You can disallow the use of system symbolics in filters by setting the custom property **Command.FILTER.SymbolsDisabled** to **TRUE**
  - System symbolic substitution is not attempted
  - Value popup with list of available symbolics is disabled

Overview: Rexx/Java Browse Enhancements

- **Problem Statement / Need Addressed**
  - Interface to browse data sets in Rexx and Java too cumbersome
    - **PARM(NP SA)** or **PARM(NP SJA)** requires user to manage allocations of individual data sets
    - The number of concurrent allocations for jobs with many data sets can result in other problems (such as below-the-line storage shortages)
- **Solution**
  - New ISFBROWSE command to browse data
  - Most ISFBROWSE features extend to existing ISFLOG command
- **Benefit / Value**
  - Easier to look at spool data sets and syslog/operlog from Rexx/Java
  - SDSF manages all allocation/unallocation and OPEN/CLOSE calls

ISFBROWSE command (REXX): Rexx/Java Browse Enhancements

ISFBROWSE *panel_name* TOKEN(*token*) (*options*

- ***panel_name*** is the panel name
  - DA, ST, I, O, H, and CK are supported
- ***token*** is the row token returned by ISFEXEC/ISFACT
  - Multiple tokens or stem variables are not supported
- ***options***
  - **JCL** option
    - Browse the JCL for the job
  - **CCASIS** option
    - ASA carriage control is not added to records with no carriage control

ISFBROWSE command (REXX): Rexx/Java Browse Enhancements

ISFBROWSE *panel_name* TOKEN(*token*) (*options*

- ***options***
  - **NOCLOSE** option
    - Data sets are not closed/unallocated unless necessary. Subsequent requests for the same row do not need to redo the open/allocation.
    - All datasets/unallocations are done when EOF is reached, or a call without NOCLOSE is done
  - **VERBOSE** option
    - Additional diagnostic messages are returned in **isfmsg2.**

REXX variables (Input): Rexx/Java Browse Enhancements

- **isflinelim** – maximum number of lines to return on this request
- **isfstartlinetoken** – represents first line to be returned
  - If null or unspecified, top-of-file is used
- **isfscrolltype** – represents any scrolling action to be done
  - **UP/DOWN** – scroll up or down
  - **NEXT/PREV** – scroll up and down by data sets
  - **FINDNEXT/FINDPREV** – find next or previous instance of a string
  - **TOP** – go to the top
  - **BOTTOM** – go to **isflinelim** lines from the bottom
- **isfscroll** – the number of lines to scroll (UP/DOWN) or number of data sets to scroll (NEXT/PREV)
  - All scrolling is relative to line passed in **isfstartlinetoken**

REXX variables (Input): Rexx/Java Browse Enhancements

- **isffind –** indicates a string to be located when FINDNEXT or FINDPREV
  - Case-insensitive search is done
- **isffindlim –** maximum number of lines to search
- **isffindstartcol -**
  - Starting column for find string
  - Minimum start column if isffindendcol is specified
- **isffindendcol –** Maximum end column for find string

REXX variables (Output): Rexx/Java Browse Enhancements

- **isfline**. stem variable – the output lines
- **isfmsg**, **isfmsg2**. - messages
- **isffirstlinetoken** – token representing first line
  - If null, indicates TOF was reached
- **isfnextlinetoken** – token representing next line after the last one that was returned
  - If null, indicates EOF was reached
- **isffirstlinedsid**, **isflastlinedsid** – dsids of the datasets in which the first and last line returned are located
- **isffirstlinerecn**, **isflastlinerecn** – record number within dataset of the first and last line returned

Example: Rexx/Java Browse Enhancements

```
rc=isfcalls("on")

Address SDSF "ISFEXEC ST"                          /* Access status panel */

do ix=1 to JNAME.0                                 /* For all jobs */

   if JNAME.ix = "D96CLW1" then                    /* Examine job name */
      do
         isflinelim = 100                          /* Return 100 lines at a time */
         do until isfnextlinetoken=''              /* Loop until EOF reached */
            Address SDSF "ISFBROWSE ST TOKEN('"token.ix"')"      /* Read lines */
            do jx=1 to isfline.0
               say isfline.jx                      /* Display lines */
            end

            isfstartlinetoken = isfnextlinetoken   /* Set next read origin */

         end
      end

end

rc = isfcalls("off")
```

Java changes: Rexx/Java Browse Enhancements

- New **browse()** and **browseJCL()** methods added (in addition to existing browseAllocate() method) for ISFActive, ISFStatus, etc.
- New methods to define browse parameters in ISFRequestSettings
  - **addISFStartLineToken(), addISFScroll(), addISFScrollType(), addISFFind(), addISFFindLim(),** etc.
- New **getLineResults()** method added to ISFRequestResults to get to line information
  - **getLineRecordList()** to obtain list of output lines
  - Other properties such as **getFirstLineToken(), getNextLineToken()**, etc. to obtain additiional information
- See Javadoc for details.

ISFLOG Enhancements: Rexx/Java Browse Enhancements

- New options on ISFLOG
  - **ISFLOG READ TYPE(SYSLOG|OPERLOG) (*options***

- **WTOR** option – returns all WTORs in isfwtor. stem variable
- **CCASIS** option – returns all records without carriage control

REXX variables (Input): Rexx/Java Browse Enhancements

- **isflinelim** – maximum number of lines to return on this request
- **isfstartlinetoken** – represents first line to be returned
- **isfscrolltype** – represents any scrolling action to be done
  - **UP/DOWN** – scroll up or down
  - **FINDNEXT/FINDPREV** – find next or previous instance of a string
  - **TOP** – go to the top
  - **BOTTOM** – go to isflinelim lines from the bottom
- **isfscroll** – the number of lines to scroll (UP/DOWN) or number of data sets to scroll (NEXT/PREV)
    - All scrolling is relative to line passed in isfstartlinetoken

REXX variables (Input): Rexx/Java Browse Enhancements

- **isffind –** indicates a string to be located when FINDNEXT or FINDPREV
  - Case-insensitive search is done
- **isffindlim** – maximum number of lines to search
- **isffindstartcol -**
  - Starting column for find string
  - Minimum start column if isffindendcol is specified
- **isffindendcol –** Maximum end column for find string
- **isflogstartdate**, **isflogstarttime**, **isflogstopdate**, **isflogstoptime**
  - Restrict the range of records within the log to be processed

REXX variables (Output): Rexx/Java Browse Enhancements

- Output
  - **isfline.** stem variable – the output lines
  - **isfwtor.** stem variable – list of WTORs if WTOR option was specified
  - **isfmsg**, **isfmsg2.** - messages
  - **isffirstlinetoken** – token representing first line
    - If null, indicates TOF was reached
  - **isfnextlinetoken** – token representing next line after the last one that was returned
    - If null, indicates EOF was reached
  - **isffirstlinedate**, **isffirstlinetime** – date and time associated with the first line returned
  - **isflastlinedate**, **isflastlinetime** – date and time associated with the last line returned

Java changes: Rexx/Java Browse Enhancements

- Use **readOperlog()** or **readSyslog()** method to obtain log data (as you do today)
- New methods to define new parameters in ISFRequestSettings
  - **addISFStartLineToken(), addISFScroll(), addISFScrollType(), addISFFind(), addISFFindLim()**, etc.
- New **getLineResults()** method added to ISFRequestResults to get to line information
  - Can use **getLineRecordList()** to obtain list of output lines
  - Other properties such as **getFirstLineToken(), getNextLineToken(),** etc. to obtain additiional information
- See Javadoc for details.

Overview: Security Assist

- **Problem Statement / Need Addressed**
  - Difficult to determine why a user cannot access functions in SDSF
  - TRACE facility potentially exposes too much information to end user
- **Solution**
  - New SECTRACE facility added
  - Documents the results of security decisions (SAF and non-SAF) made by SDSF
  - Intent is to diagnose why access to panels, commands, or actions is denied
  - Row-by-row decisions (such as MLS checks) are not traced
- **Benefit / Value**
  - Access issues easier to diagnose

SET SECTRACE: Security Assist

- The new **SET SECTRACE** command is used to control whether new security trace message are to be issued
- The syntax is as follows:

  SET SECTRACE ON | ULOG | WTP | OFF | ?

  where

  ON or **ULOG** indicates that messages are placed in ULOG

  WTP indicates that messages are to be issued as write-to-programmer

  OFF indicates that no messages are to be issued

  ? indicates the current value is to be queried

- The value of SET SECTRACE is not saved in the ISPF profile across SDSF sessions.

SECTRACE start option: Security Assist

- **SECTRACE(ON)**, **SECTRACE(ULOG)**, or **SECTRACE(WTP)** can be specified as a start option when invoking SDSF
  - Access to checks done before command processing is available
    - Group assignment
    - Main panel construction
  - Can be turned off by SET SECTRACE once SDSF is up.

Rexx/Java: Security Assist

- The new **isfsectrace** variable is used to control whether new security trace message are to be issued
- Values are slightly different in meaning

  ON indicates that messages are placed in isfmsg2.

  ULOG indicates that messages are placed in isfulog.

  WTP indicates that messages are to be issued as write-to-programmer

  OFF indicates that no messages are to be issued

- Early tracing (group assignment etc.) is always done in Rexx/Java when ON, ULOG, or WTP is specified

What does SECTRACE trace?: Security Assist

- USER, GROUP, PROC, and terminal name
  - Message issued when SECTRACE is turned on
  - This information factors into subsequent decisions
  - Will be reissued every time SET SECTRACE ON/ULOG/WTP is entered
- SAF decisions
  - Resource, access requested, and result are issued for non-MLS security checks
- SAF overrides by ISFUSER exit
- Non-SAF decisions based on ISFPARMS or ISFUSER
  - Usually fallback scenarios when SAF profile doesn't exist

Examples of SECTRACE messages: Security Assist

- **ISF050I USER=D96CLW1 GROUP=ISFSPROG PROC=SDSF31EJ TERMINAL=LOCALC11**
- **ISF051I SAF Access allowed SAFRC=0 ACCESS=READ CLASS=SDSF RESOURCE=ISFCMD.DSP.STATUS.JES2**
- **ISF051I SAF Access denied SAFRC=8 ACCESS=UPDATE CLASS=SDSF RESOURCE=ISFATTR.JOB.CLASS**
- **ISF051I SAF No decision SAFRC=4 ACCESS=UPDATE CLASS=SDSF RESOURCE=ISFATTR.JOB.PRTY**
- **ISF058I COLUMN=Prty Access allowed USERLEVEL=7 REQLEVEL=3**
- **ISF053I COMMAND=PREFIX Access allowed**
- **ISF055I ACTION=H Access denied USERLEVEL=7 REQLEVEL=3 BPXAS S0000027 RSN=08 job name in exclude list**
- **ISF057I GROUP=ISFSPROG Access allowed USERAUTH=OPER,ACCT,JCL REQAUTH=OPER,ACCT,JCL**
- **ISF051I SAF Access allowed SAFRC=0 ACCESS=READ CLASS=JESSPOOL RESOURCE=SYSA1N.D96CLW1.D96CLW1.T0000036.D0000002.JESMSGLG**

Migration & Coexistence Considerations

- Sharing SDSF V1R13 Server Parms with lower releases of SDSF
  - If you are sharing ISFPRMxx or ISPF profiles with pre-SDSF 1.13 systems you must install the toleration PTFs associated with APARs **PM37714**, **PM78102,** and **PM87996**

Migration & Coexistence Considerations

- Other migration & coexistence considerations:
  - GROUP column on JES2 JC display by default is inserted after STATUS column
    - May affect applications that rely on columns at specific locations on screen
    - ISFPARMS can change the default order/width if required.
  - If you create a filter containing a system symbolic, it may be deleted if you later use a pre-z/OS V2R1 version of SDSF.

Presentation Summary

- Describe the SDSF support new for release V2R1:
  - 64-bit support
  - 8-character job class
  - Print enhancements
  - Sort enhancements
  - JESPlex scoping
  - System symbolics support
  - REXX/Java enhancements (BROWSE and LOG)
  - Security Assist

Appendix

- SDSF Operator and Customization, SA22-7670-15
- SDSF REXXHELP command
  - Contains SDSF/REXX usage, syntax, and examples
- SDSF Javadoc
  - Contains all SDSF Java documentation
- SDSF SEARCH command
  - Searches SDSF help system for word or phrase