New SDSF Functions in z/OS 3.1 - Preview

Session 52515

March 7, 2023

Rob Scott

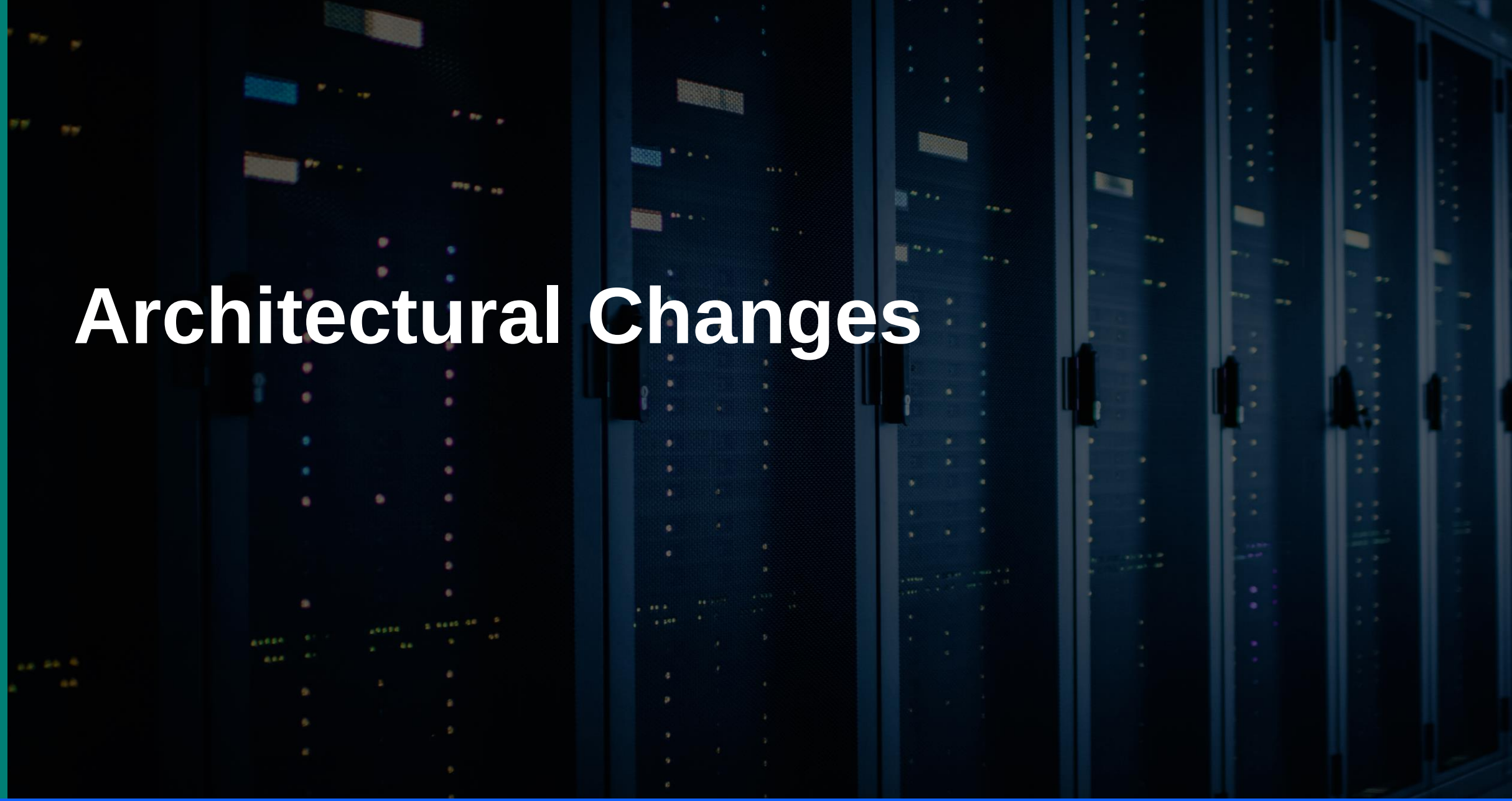
Rocket Software

New SDSF Functions in z/OS 3.1 - Preview

Rob Scott
Rocket Software
rscott@rs.com

Agenda

- Architectural Changes
- SDSF Event Log
- Module Fetch Monitor
- RACF Panels
- MEM Enhancements
- Useability Enhancements



Architectural Changes

Architectural Changes

- New SDSF concept of a “feature”
  - Optional component of SDSF that can be configured via new FEATURE and FEATENT statements in ISFPRMxx
  - Certain SDSF panels can depend on a specific feature being active
  - FEATURE statement
    - Initial status of the feature when SDSF server starts
    - Resource limits for any data collection
    - Level of detail maintained
  - FEATENT statements
    - One or more optional filters for the data collected
  - Features can be dynamically started and stopped using operator commands
  - SDSF for z/OS ships with two features available
    - Module Fetch Monitor
    - Event Log

SDSF Event Log

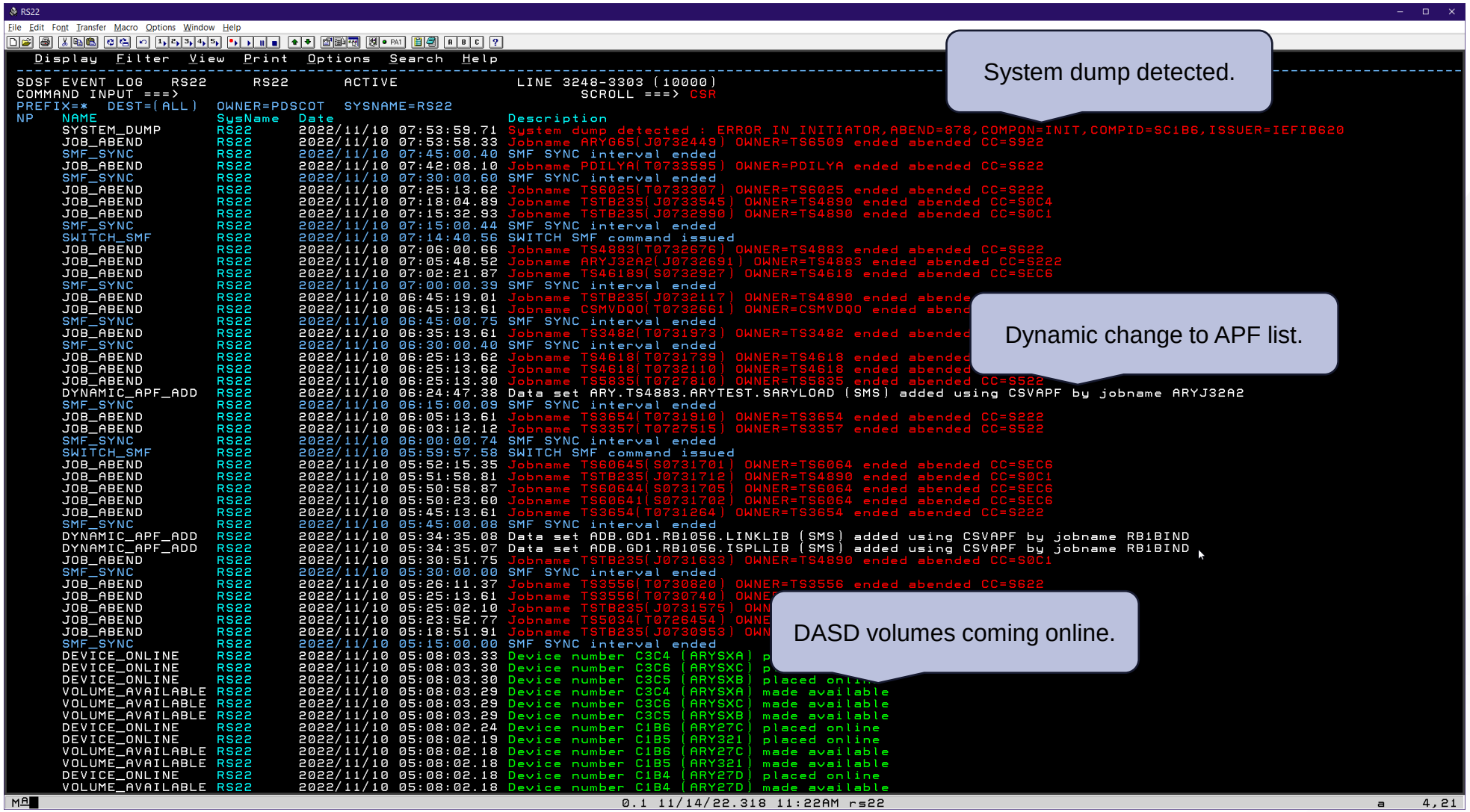
Event Log Implementation

- Enabled by new FEATURE statement in ISFPRMxx
  - Is active by default
- Captures important recent system events (default 10,000) and presents the SDSF user with a summary row for each event
- The “L” action against the row takes the user to the nearest OPERLOG record to the event timestamp
  - Can save numerous “FIND xxx PREV” activities in the LOG panel
- Data sources include :
  - ENF signals
  - SMF records
  - Dynamic exits
- SDSF assigned category and event names can be used to filter data collected on FEATENT statements in ISFPRMxx

Event Log Captured Data Examples

- System status changes
  - I/O configuration
  - WLM policy
  - Boost activity
  - SET operator commands
  - RSM detected shortages and any subsequent relief
- Dynamic updates to system resources
  - APF list
  - Link list
  - LPA
- Abnormal end to jobs
- SVC dump creation

Event Log Panel (ELOG)

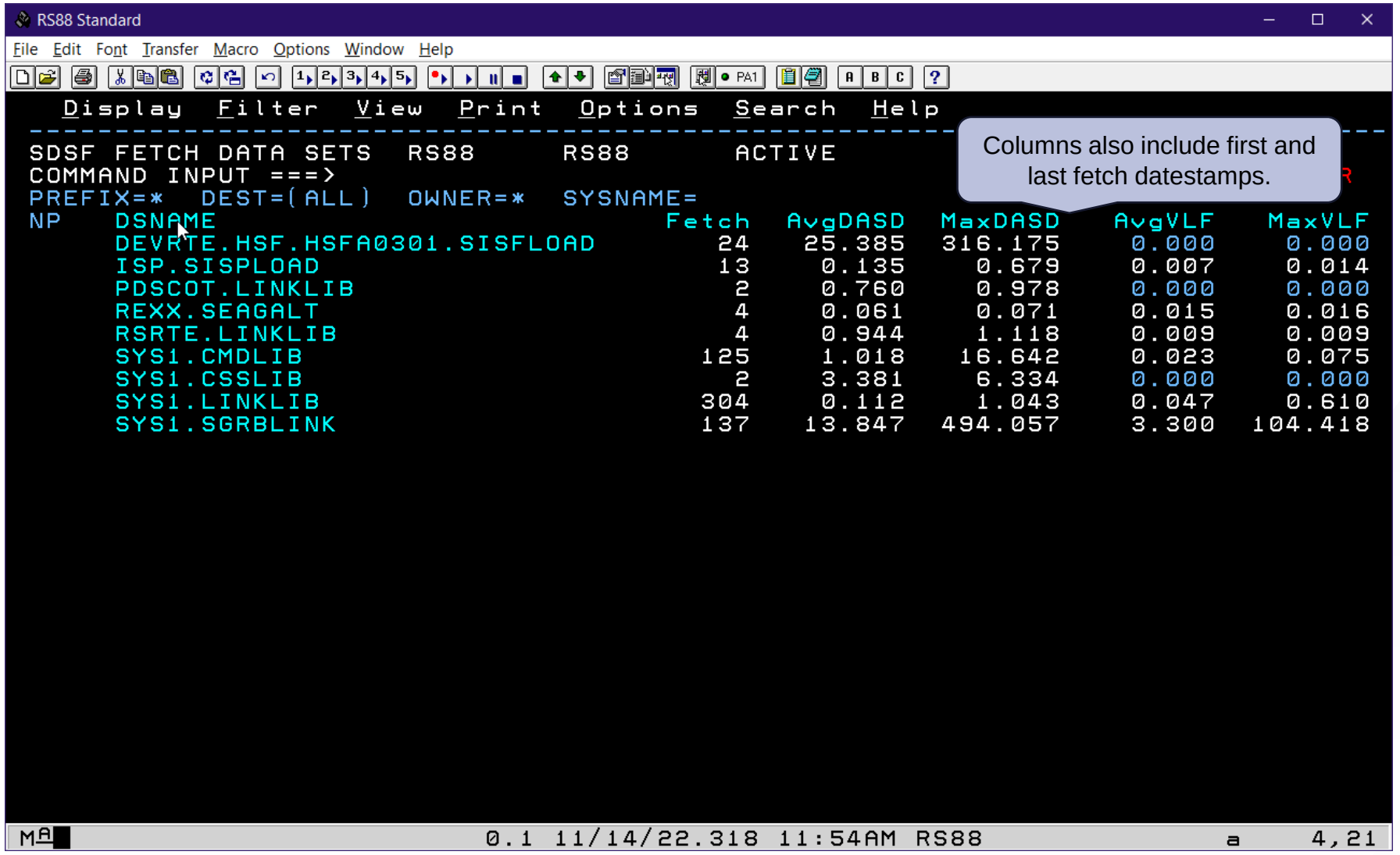


Module Fetch Monitor

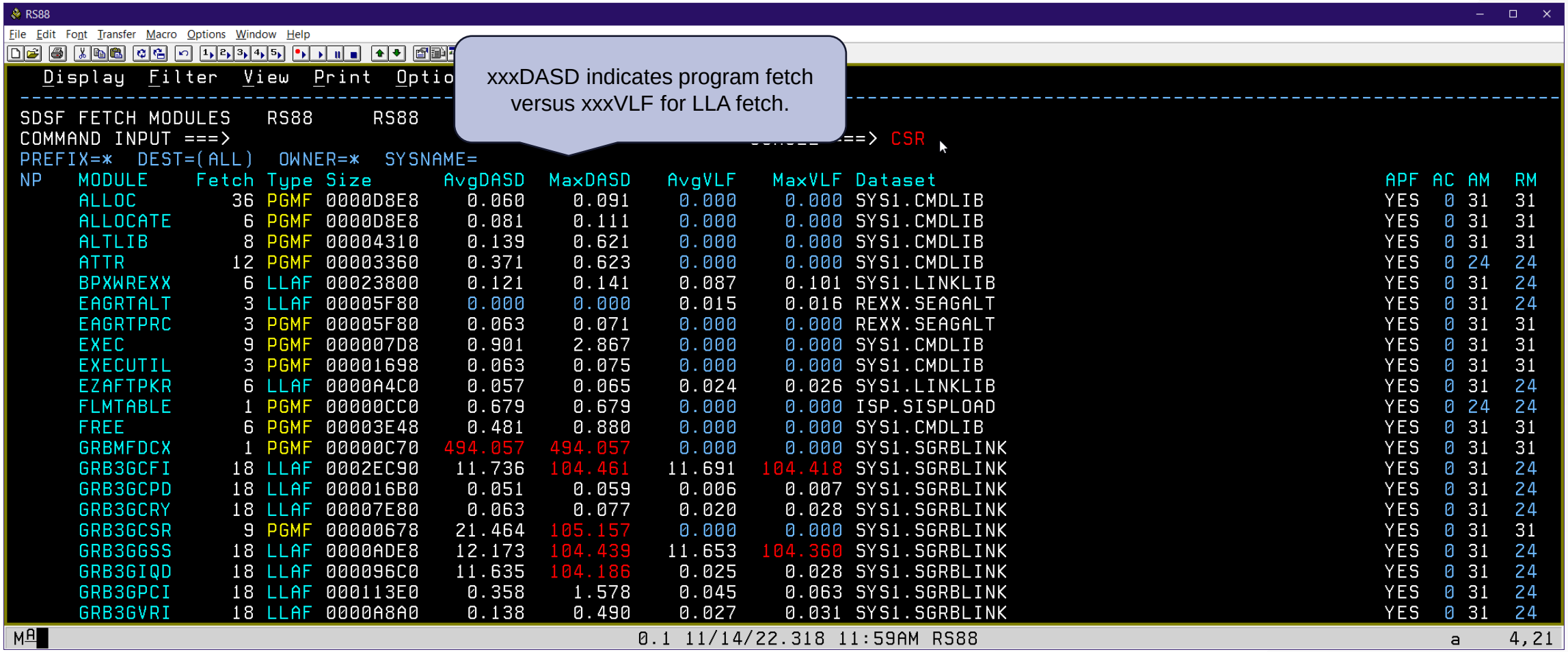
Module Fetch Monitor Implementation

- Enabled by new FEATURE statement in ISFPRMxx
  - Is inactive by default as non-trivial overhead
- Captures program fetch data from CSVFETCH and CSVLLIX1 dynamic exits
- Support 3 levels of data collection detail on FEATURE statement
  - LEVEL(1) Only collect module fetch data for data set names
  - LEVEL(2) Collect module fetch data for both data set and module name
  - LEVEL(3) Collect module fetch data for data set name, module name and job names
- FEATENT statements can filter on data set name, module name and fetch-causing jobname

Module Fetch By Data Set Name



Module Fetch By Module Name



Display Filter View Print Optio

SDSF FETCH MODULES RS88 RS88

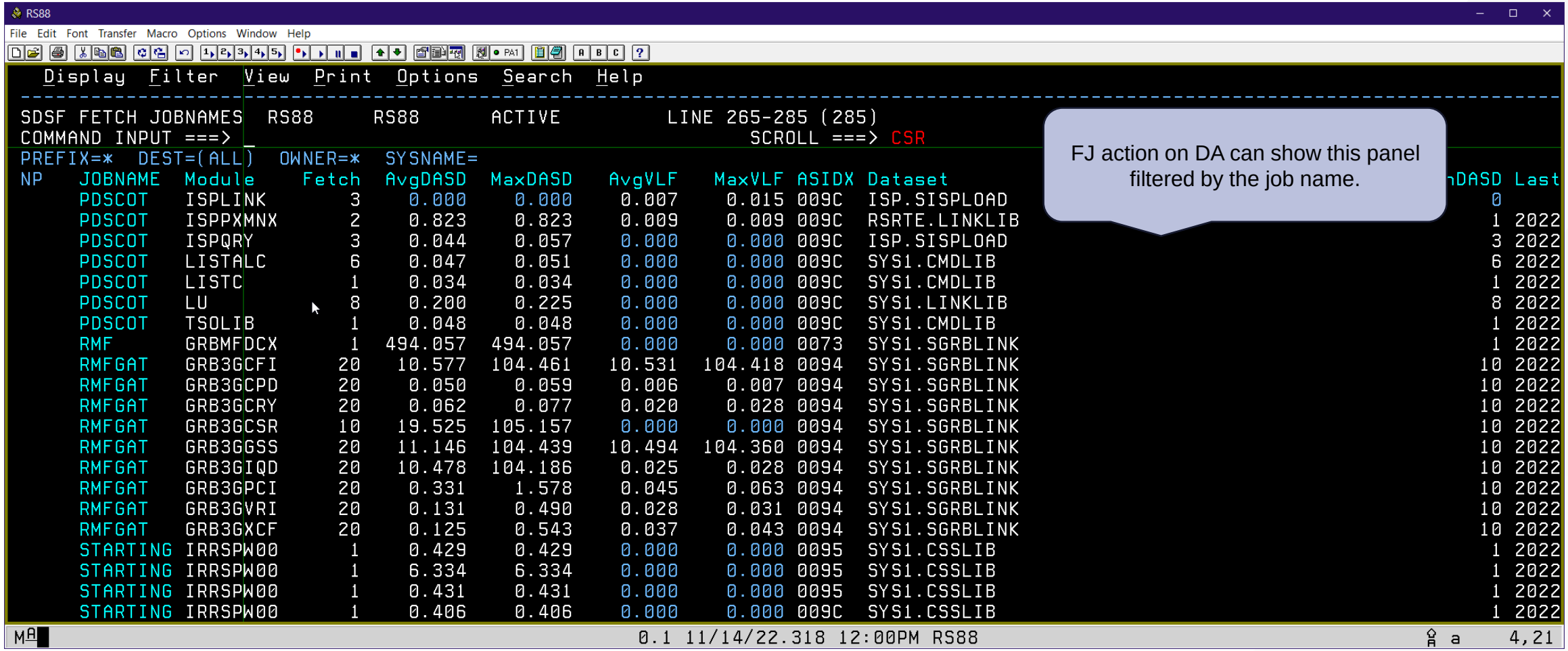
COMMAND INPUT ===>

PREFIX=* DEST=(ALL) OWNER=* SYSNAME=

NP	MODULE	Fetch	Type	Size	AvgDASD	MaxDASD	AvgVLF	MaxVLF	Dataset	APF	AC	AM	RM
	ALLOC	36	PGMF	0000D8E8	0.060	0.091	0.000	0.000	SYS1.CMDLIB	YES	0	31	31
	ALLOCATE	6	PGMF	0000D8E8	0.081	0.111	0.000	0.000	SYS1.CMDLIB	YES	0	31	31
	ALTLIB	8	PGMF	00004310	0.139	0.621	0.000	0.000	SYS1.CMDLIB	YES	0	31	31
	ATTR	12	PGMF	00003360	0.371	0.623	0.000	0.000	SYS1.CMDLIB	YES	0	24	24
	BPXWREXX	6	LLAF	00023800	0.121	0.141	0.087	0.101	SYS1.LINKLIB	YES	0	31	24
	EAGRTALT	3	LLAF	00005F80	0.000	0.000	0.015	0.016	REXX.SEAGALT	YES	0	31	24
	EAGRTPRC	3	PGMF	00005F80	0.063	0.071	0.000	0.000	REXX.SEAGALT	YES	0	31	31
	EXEC	9	PGMF	000007D8	0.901	2.867	0.000	0.000	SYS1.CMDLIB	YES	0	31	31
	EXECUTIL	3	PGMF	00001698	0.063	0.075	0.000	0.000	SYS1.CMDLIB	YES	0	31	31
	EZAFTPKR	6	LLAF	0000A4C0	0.057	0.065	0.024	0.026	SYS1.LINKLIB	YES	0	31	24
	FLMTABLE	1	PGMF	00000CC0	0.679	0.679	0.000	0.000	ISP.SISPLOAD	YES	0	24	24
	FREE	6	PGMF	00003E48	0.481	0.880	0.000	0.000	SYS1.CMDLIB	YES	0	31	31
	GRBMFDCX	1	PGMF	00000C70	494.057	494.057	0.000	0.000	SYS1.SGRBLINK	YES	0	31	31
	GRB3GCFI	18	LLAF	0002EC90	11.736	104.461	11.691	104.418	SYS1.SGRBLINK	YES	0	31	24
	GRB3GCPD	18	LLAF	000016B0	0.051	0.059	0.006	0.007	SYS1.SGRBLINK	YES	0	31	24
	GRB3GCRY	18	LLAF	00007E80	0.063	0.077	0.020	0.028	SYS1.SGRBLINK	YES	0	31	24
	GRB3GCSR	9	PGMF	00000678	21.464	105.157	0.000	0.000	SYS1.SGRBLINK	YES	0	31	31
	GRB3GGSS	18	LLAF	0000ADE8	12.173	104.439	11.653	104.360	SYS1.SGRBLINK	YES	0	31	24
	GRB3GIQD	18	LLAF	000096C0	11.635	104.186	0.025	0.028	SYS1.SGRBLINK	YES	0	31	24
	GRB3GPCI	18	LLAF	000113E0	0.358	1.578	0.045	0.063	SYS1.SGRBLINK	YES	0	31	24
	GRB3GVRI	18	LLAF	0000A8A0	0.138	0.490	0.027	0.031	SYS1.SGRBLINK	YES	0	31	24

MA 0.1 11/14/22.318 11:59AM RS88 a 4,21

Module Fetch By Job Name



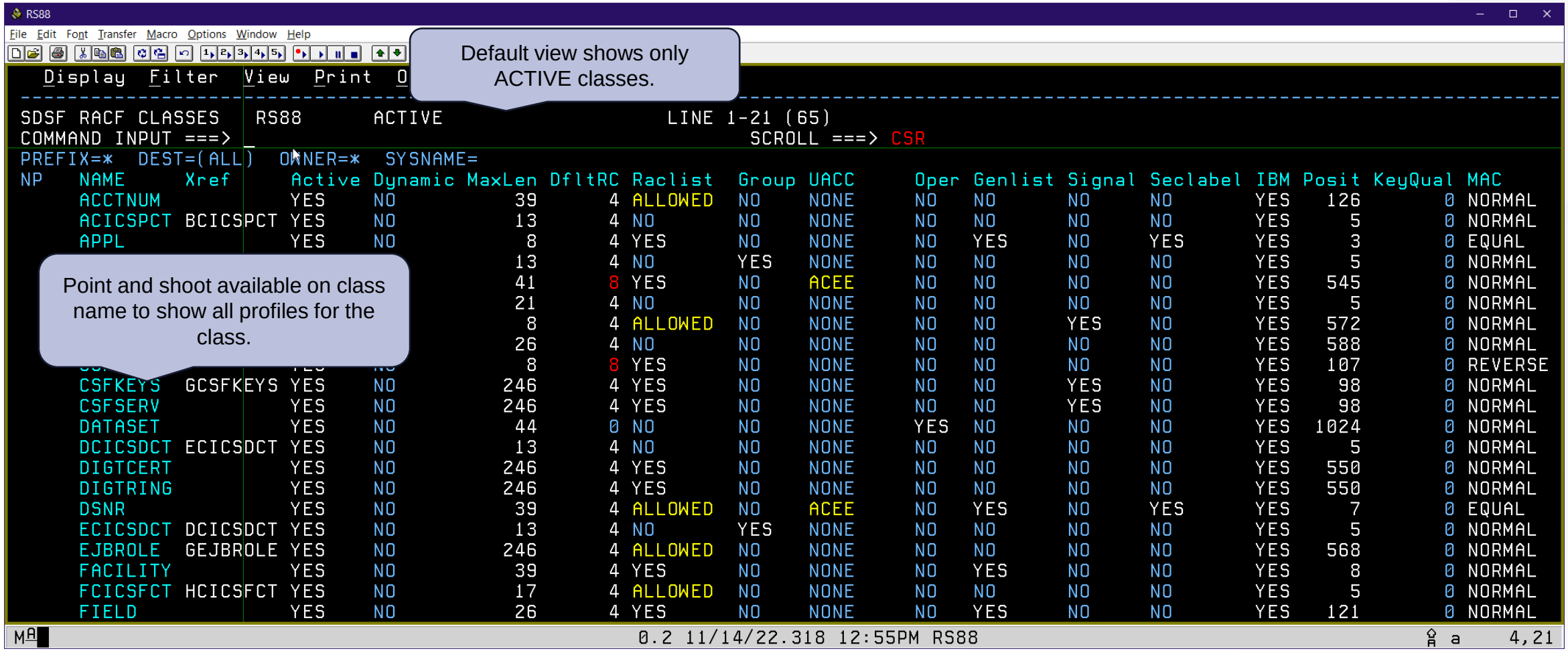
FJ action on DA can show this panel filtered by the job name.

RACF Panels

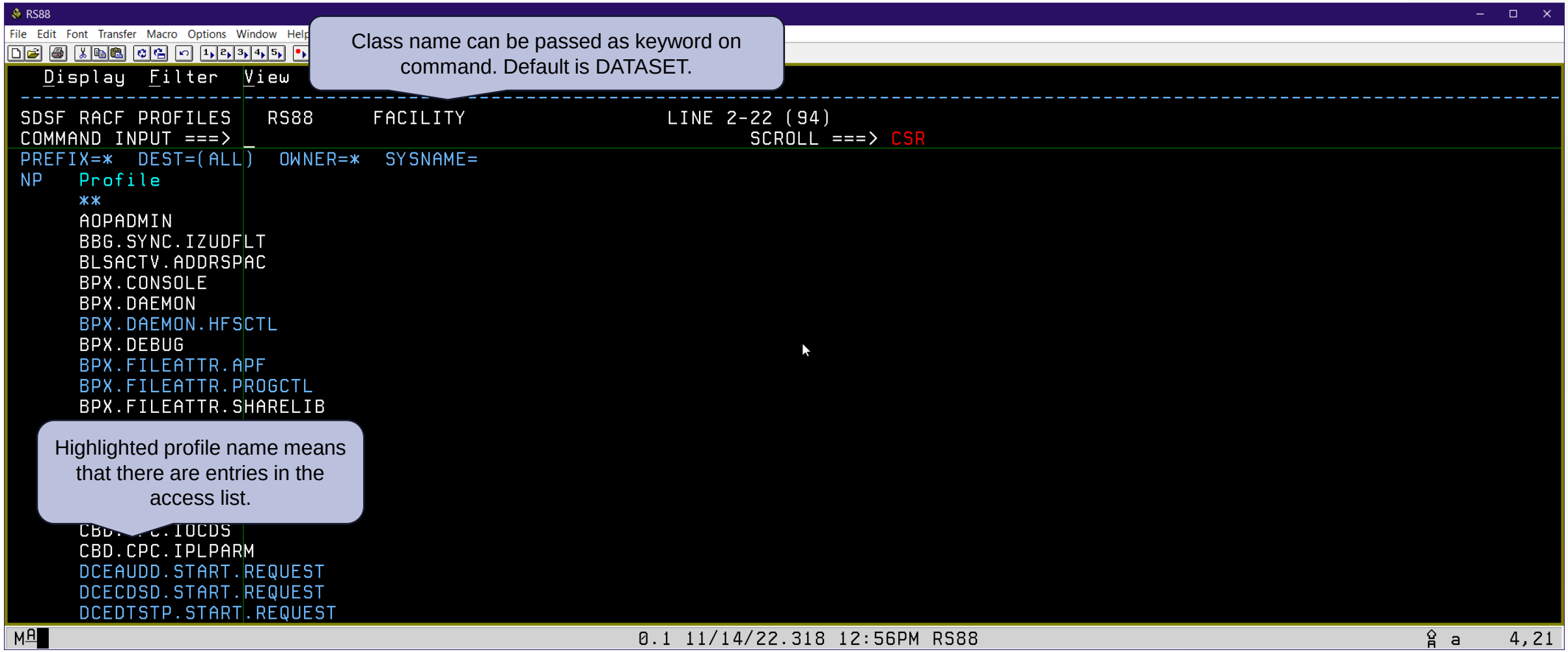
RACF Panels

- Several RACF-related panels added
  - RACF classes
  - RACF profiles
  - Access lists for specific profile
  - Connected groups list for specific userid
  - Profile “browse”
  - RACF options (SETROPTS)
- Uses IRRSEQ00 (Radmin) to collect data
  - For performance reasons, SDSF will not check the IRR.RADMIN.* profiles in the FACILITY class
  - Ability to see RACF data based purely on READ access to ISFCMD.ODSP.RACLIST.sysname in the SDSF class

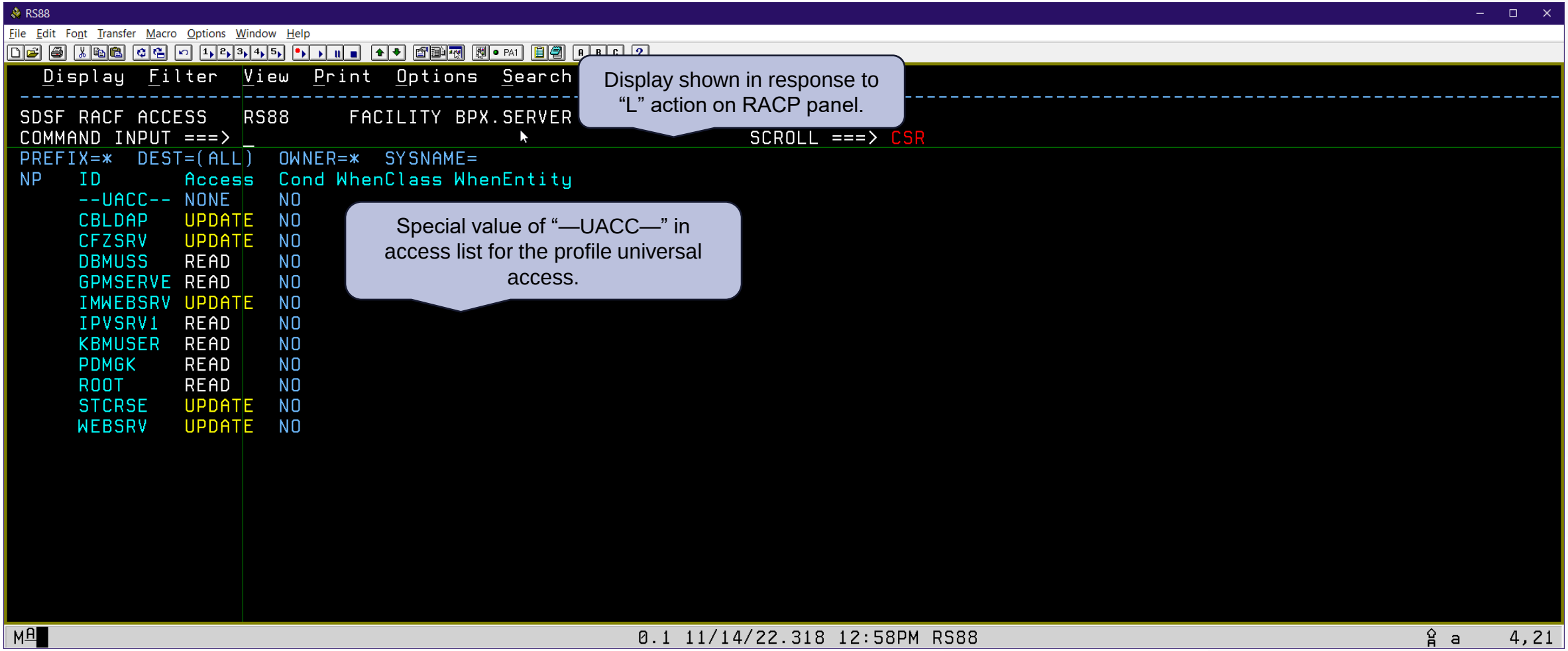
RACF Classes (RAC)



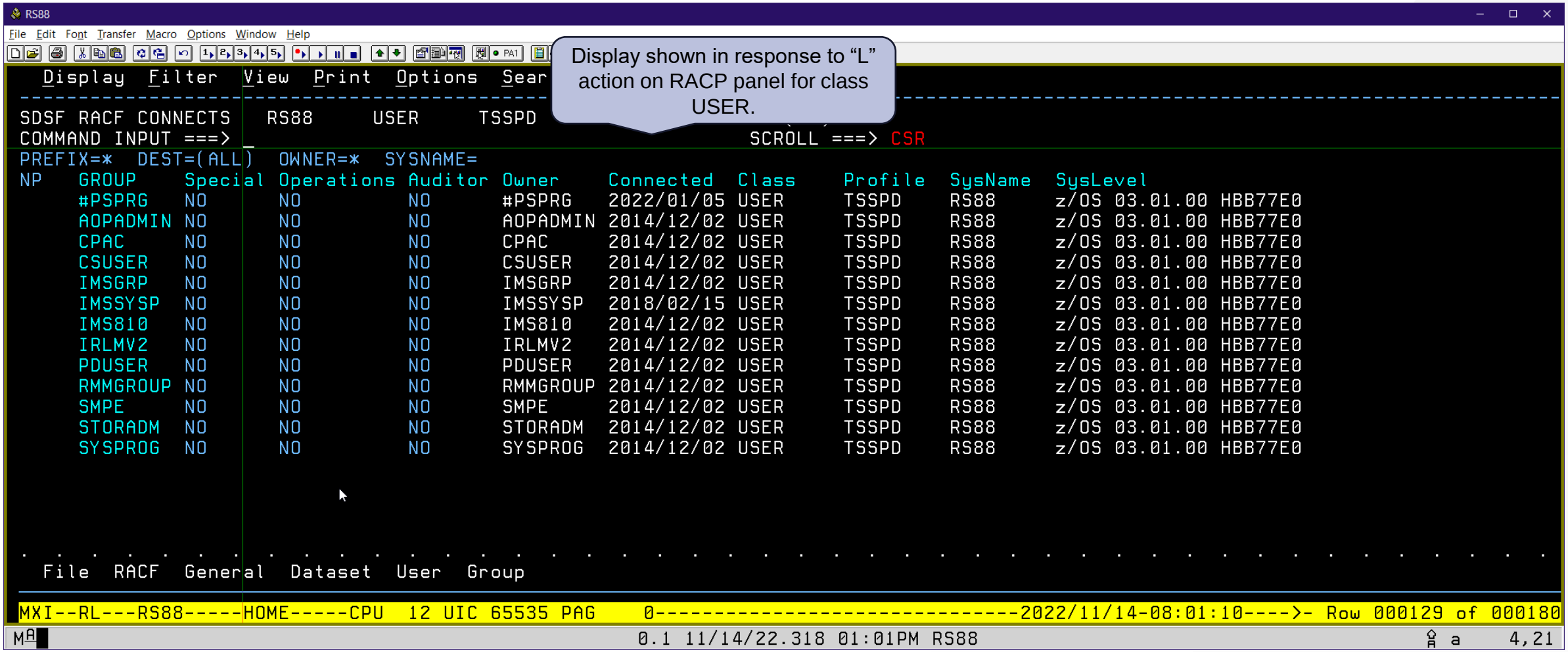
RACF Profiles (RACP)



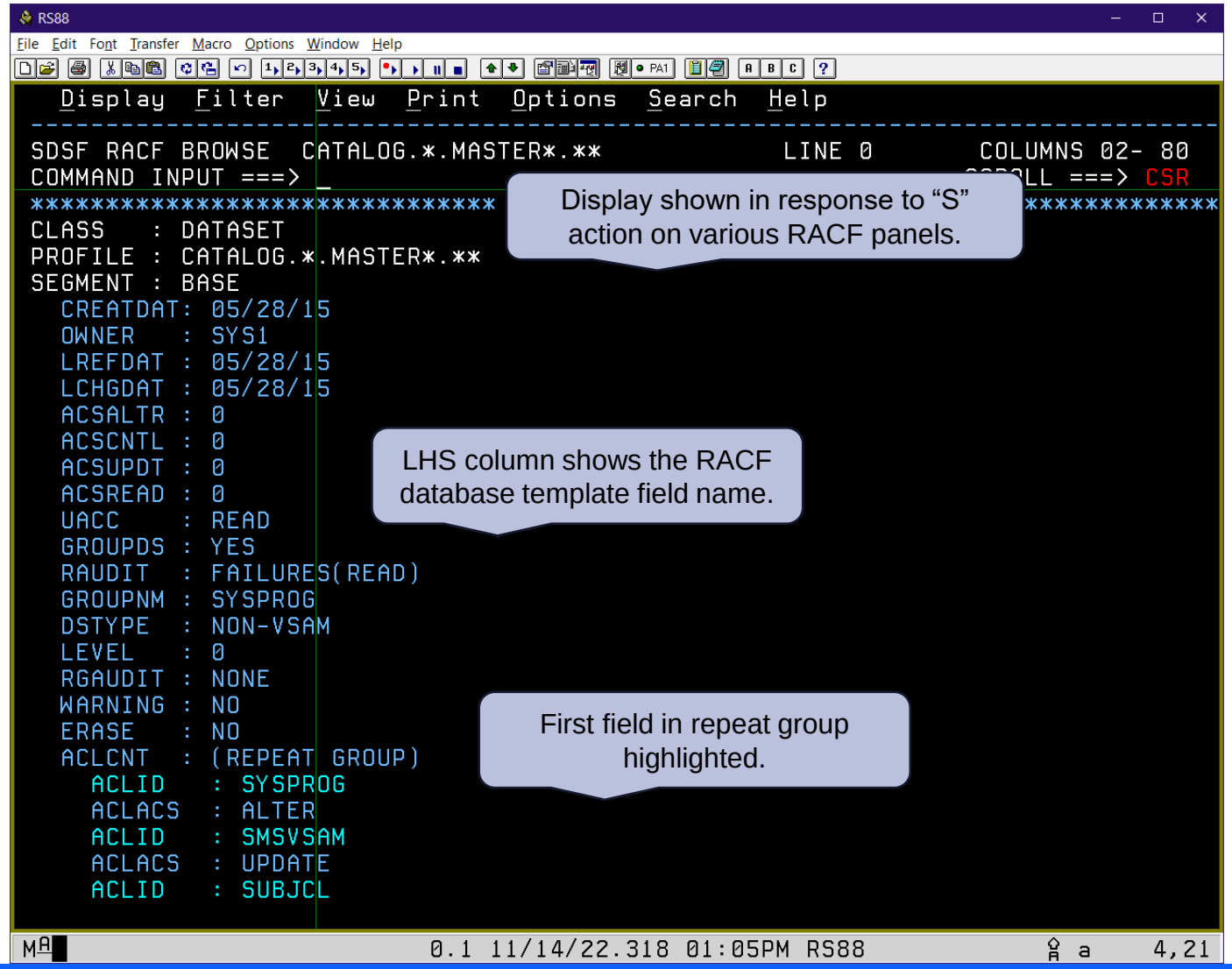
RACF Profile Access List (RACA)



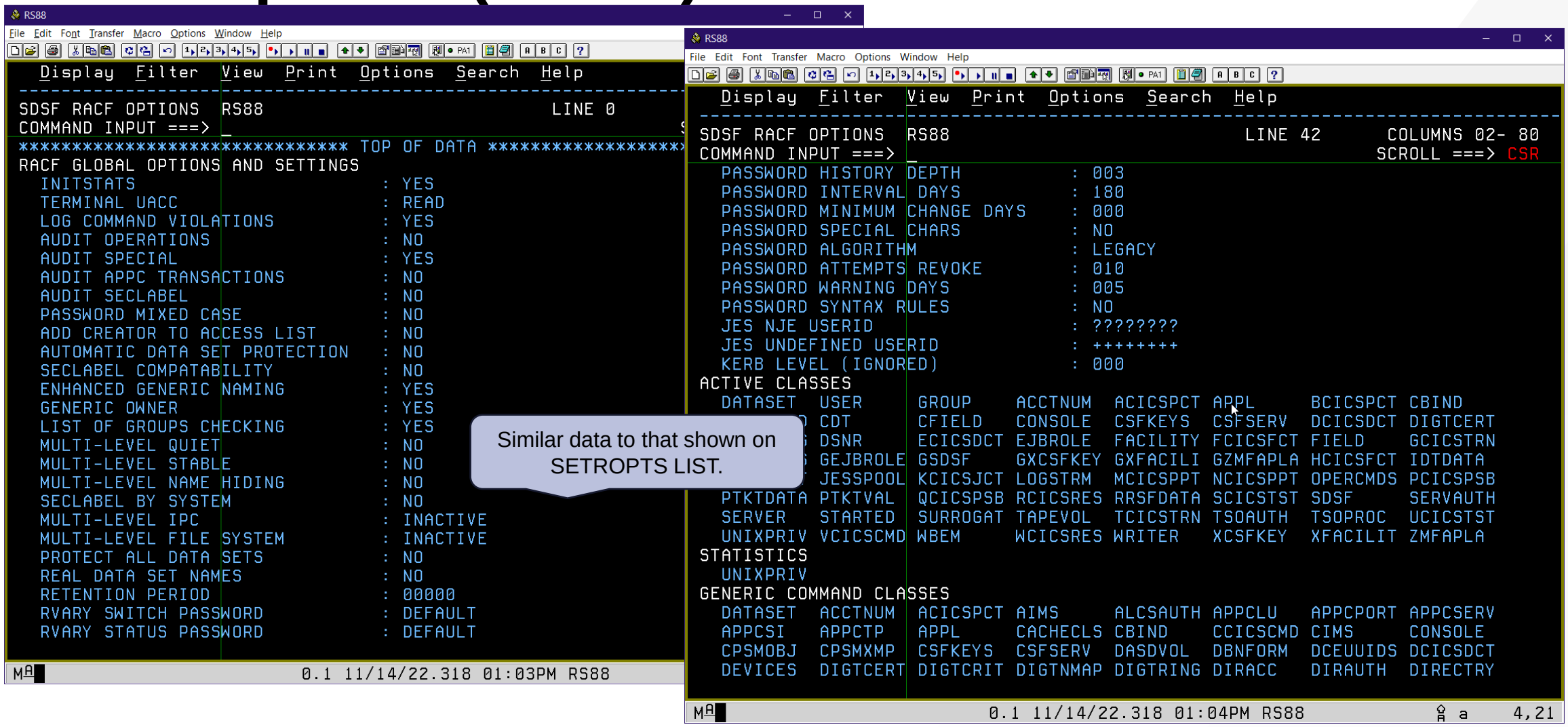
RACF Connected Groups (RACC)



RACF Profile Browse



RACF Options (RACO)

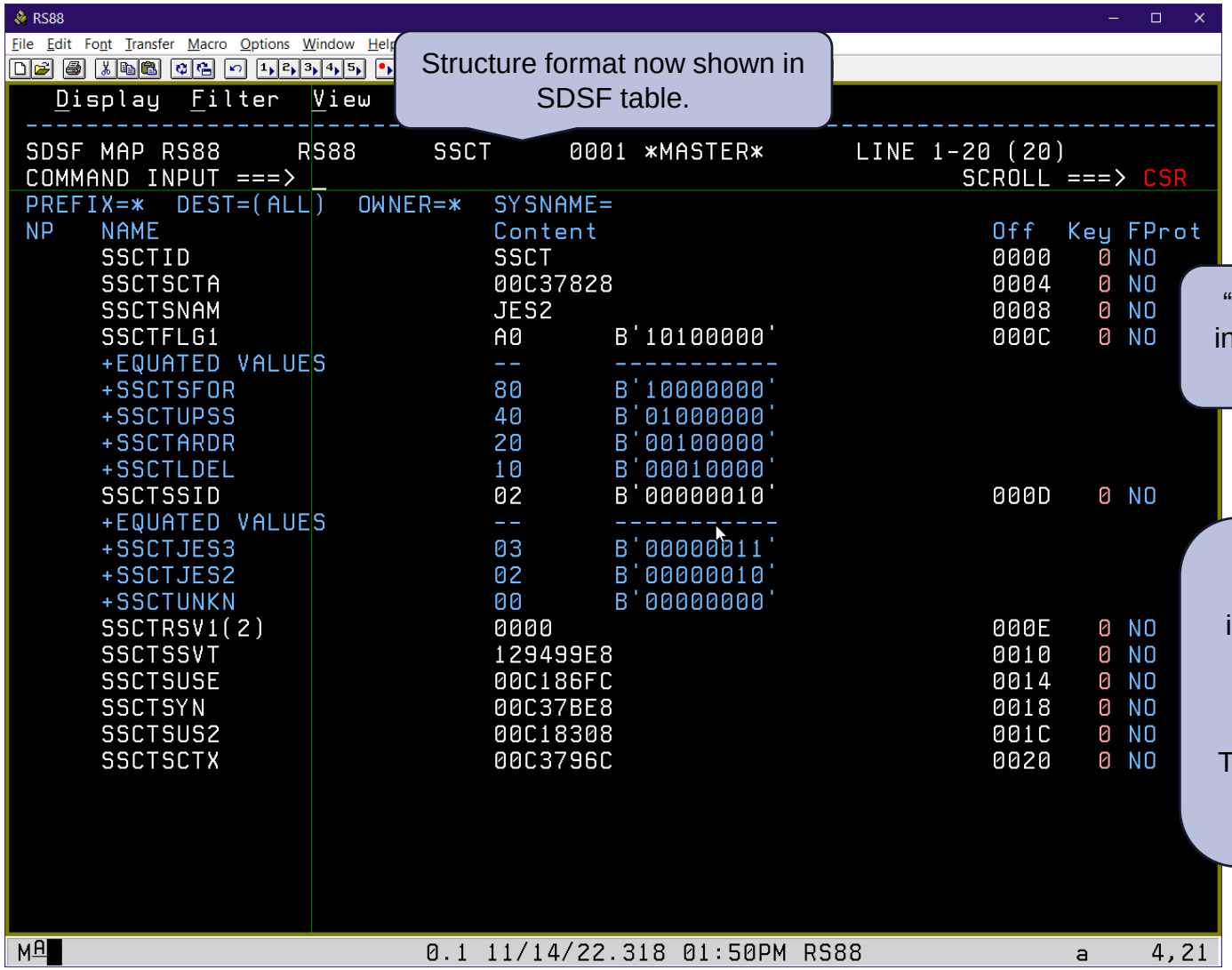


MEM Enhancements

MEM Enhancements

- Structure format now uses SDSF table display instead of browse
  - “D” and “G” actions supported on the MEM MAP panel
  - No longer need “MEMCSR” assigned to PF-Key
- Structure definitions externalized to new ISFPRMxx statements
  - MAP
  - MAPENT
  - MAPOPT
  - MAPDEF
- New “RC” action added to MEM display to perform a “run chain” to process linked lists

MEM MAP Table Format



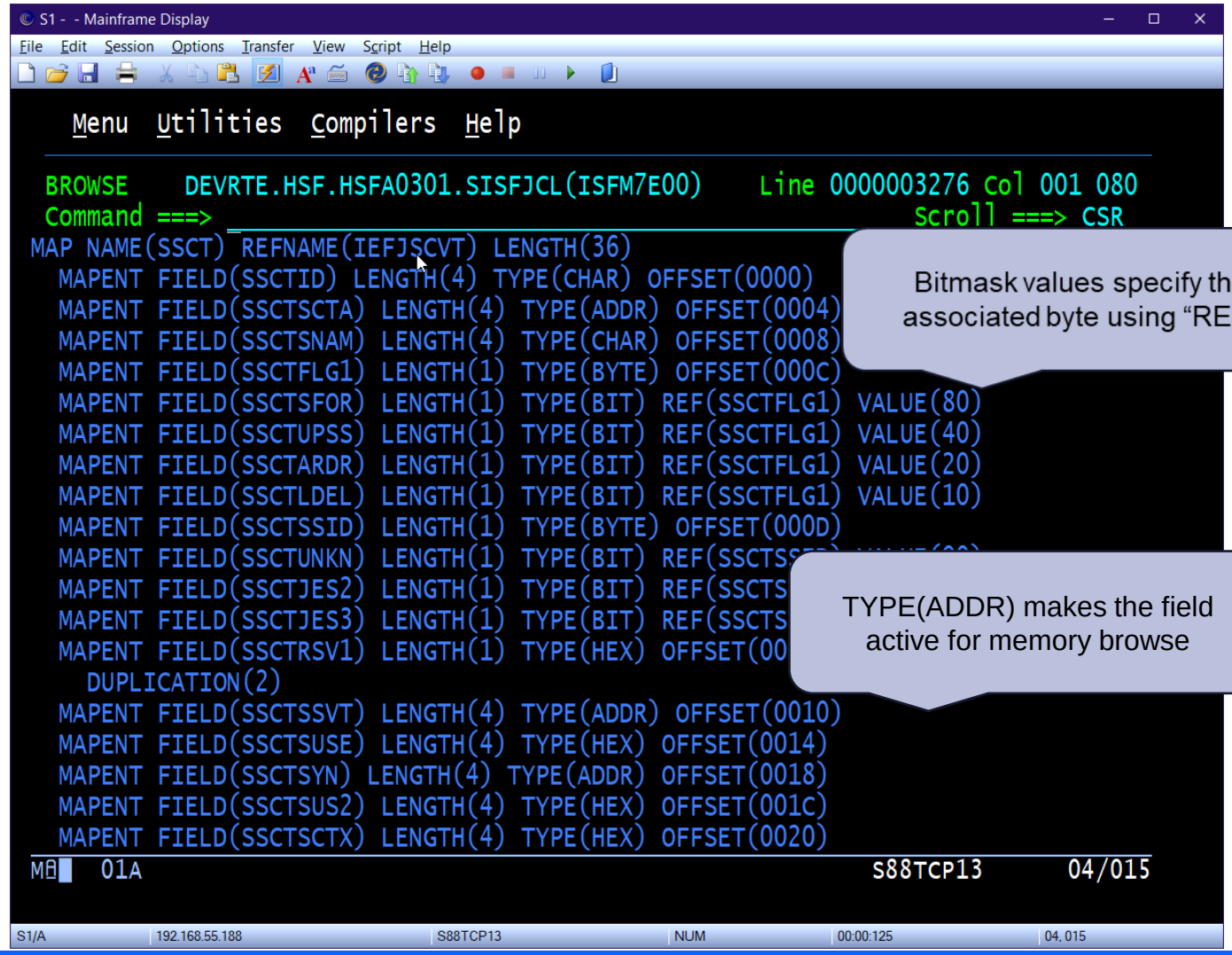
Structure format now shown in SDSF table.

"D" and "G" actions available to invoke MEM on address value in Content field.

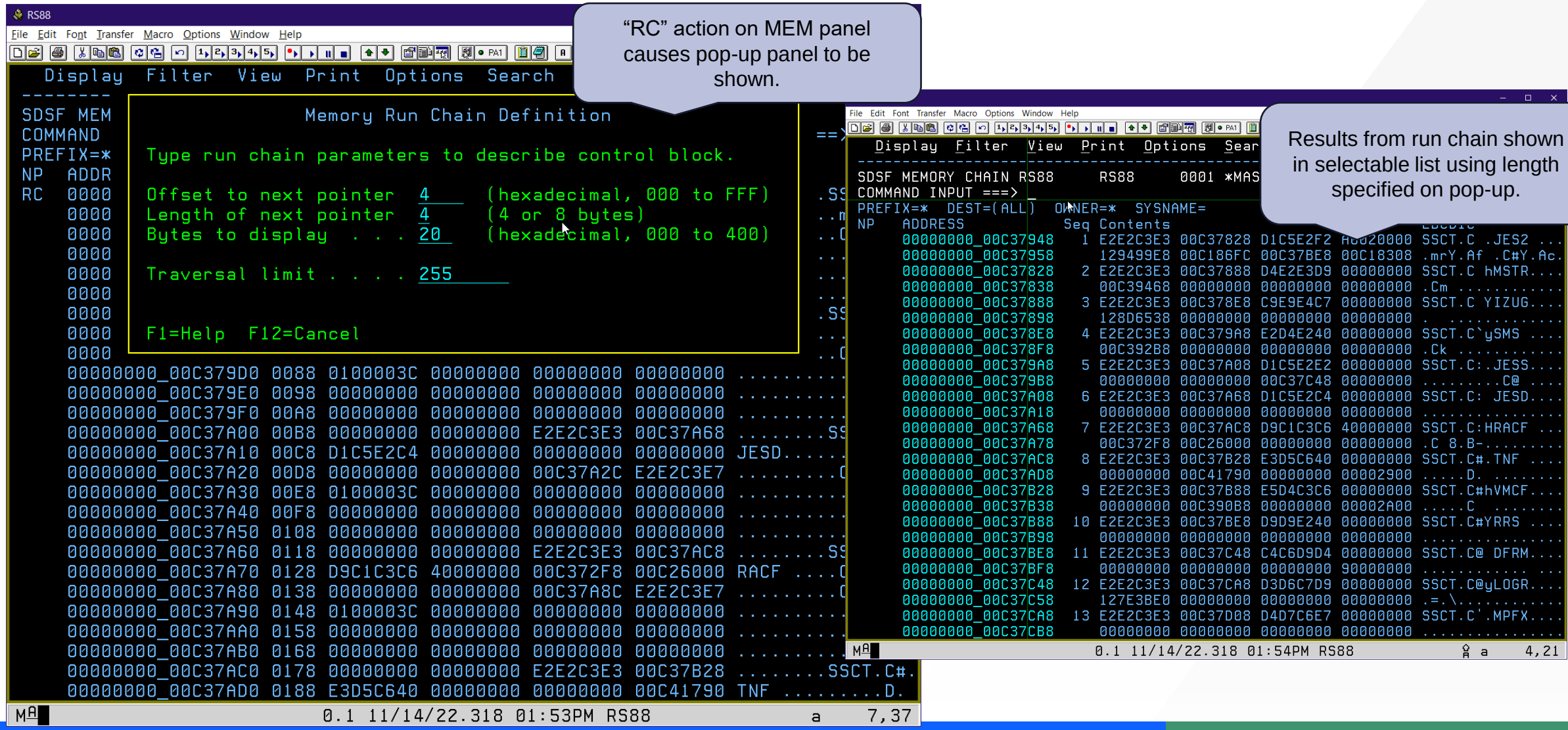
"M" action against a row will invoke MEM for the address and then format the new content inheriting the same map.

This allows user to process linked list in structure format.

Example MAP Structure Definition



MEM Run Chain

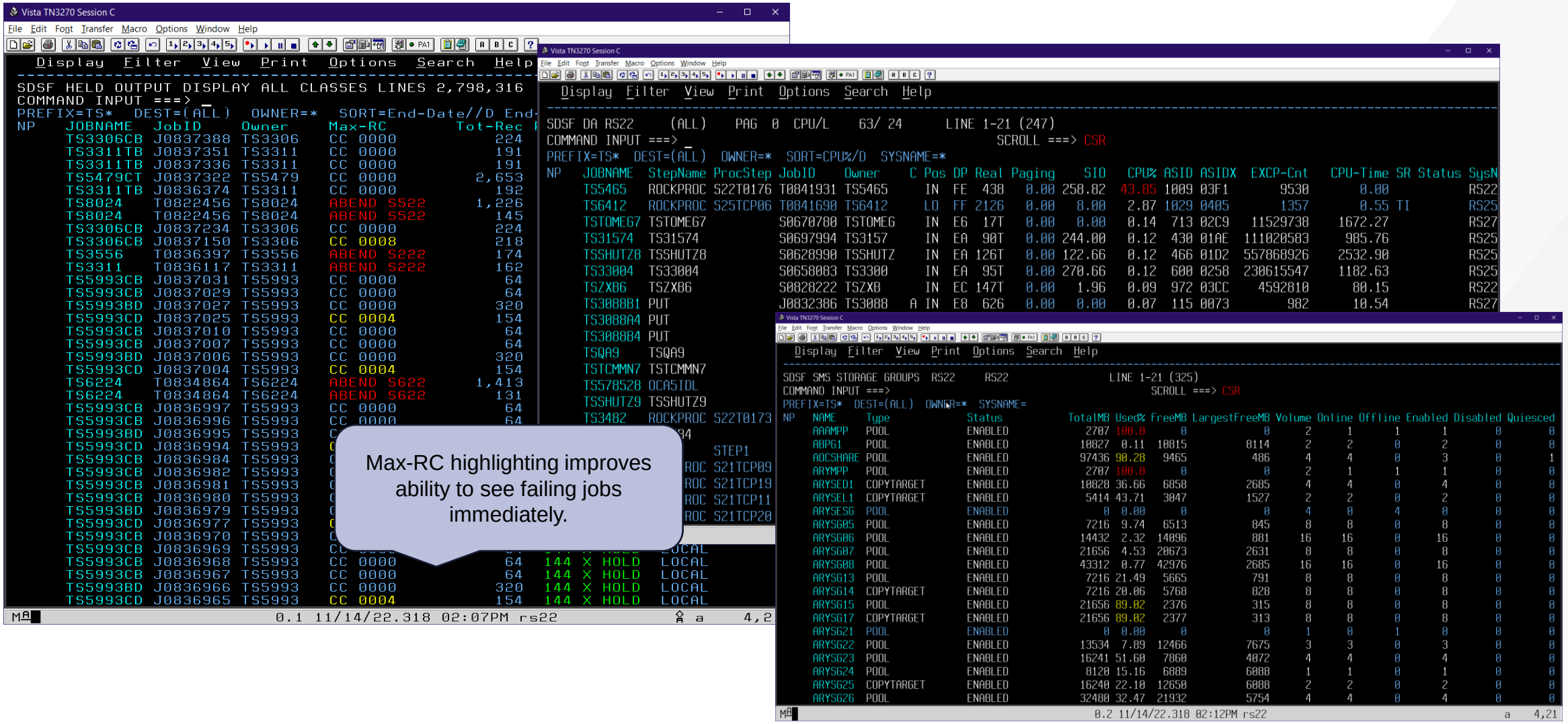


Useability Enhancements

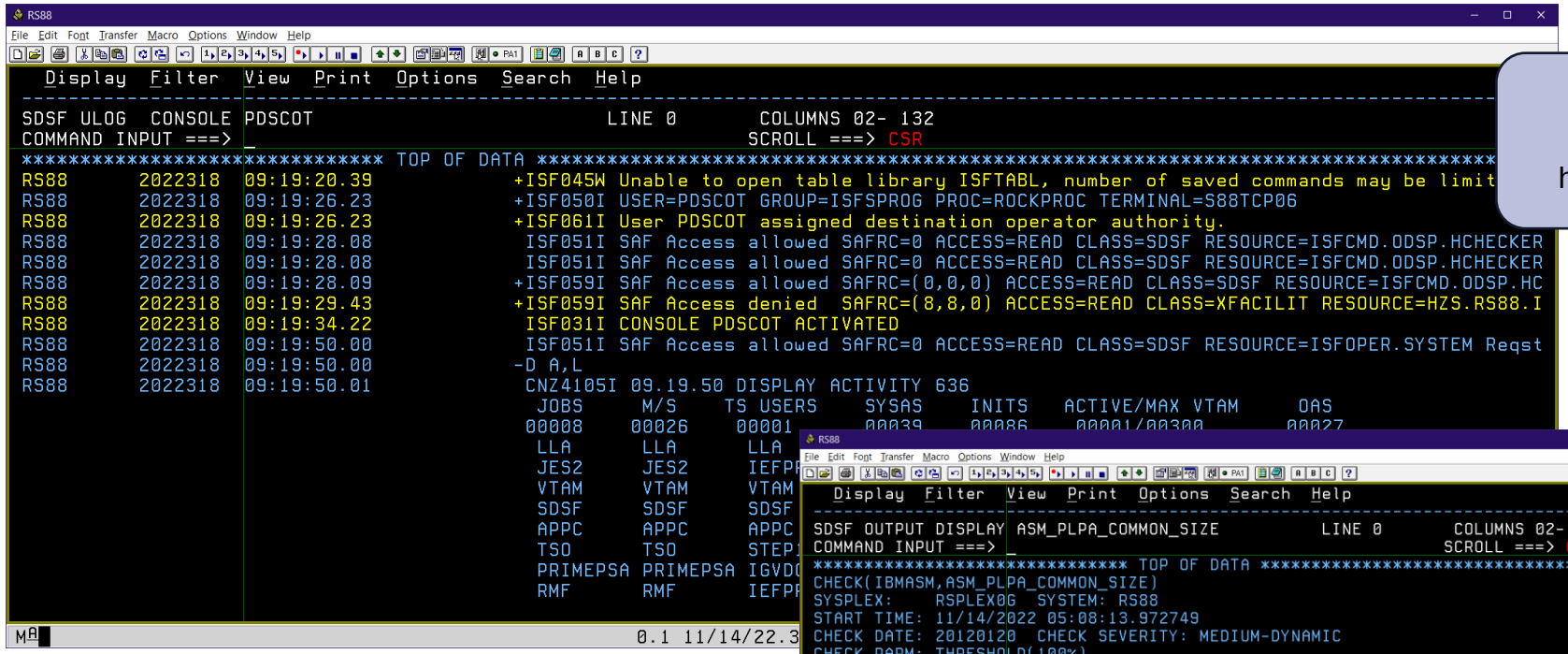
Extended Field Highlighting

- Fields on numerous panels now utilize extended highlighting
- Colour change for both character and numerical data depending on value
- Thresholds and ranges currently use internal definitions
- Examples
  - Non-zero return codes on ST and H
  - Percentage values on displays like DA and SMSG
  - Access level permitted for RACF profile
  - Number of extents for data set

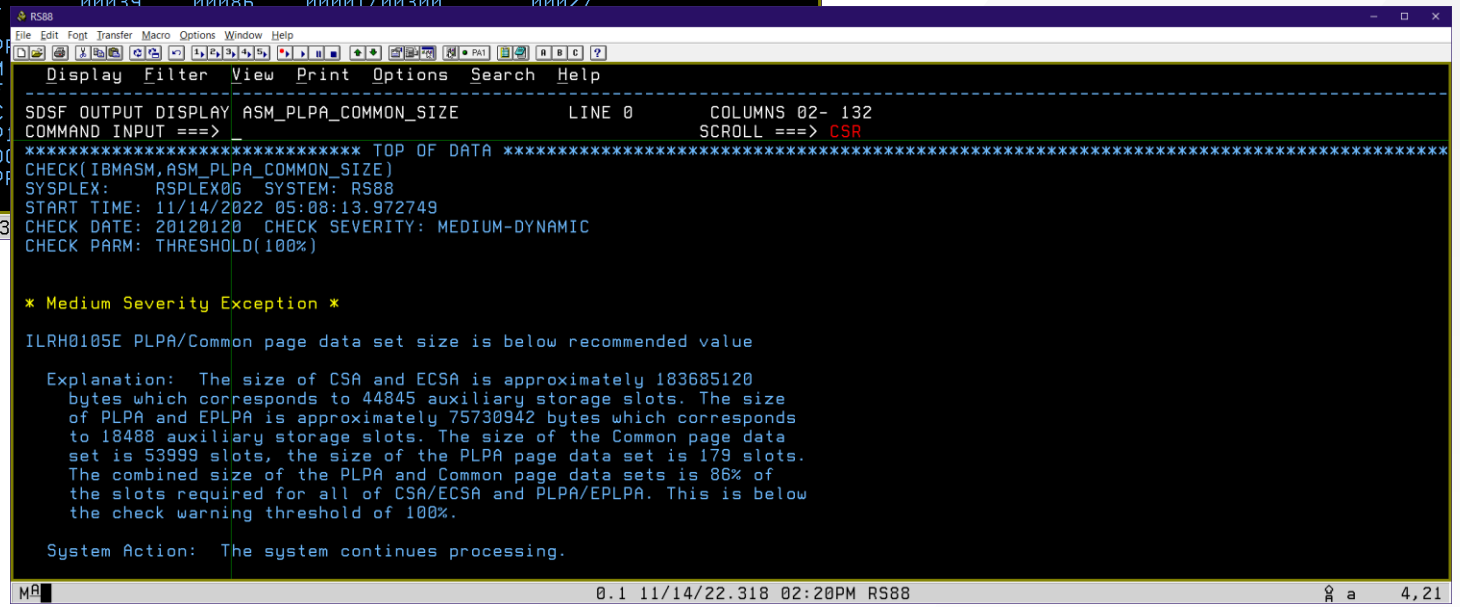
Extended Field Highlighting - Examples



Extended Field Highlighting - Examples



Enhancements to certain “browse” style panels to highlight individual records.



Summary

- New concept of SDSF feature with two supplied components
  - Event Log
  - Module Fetch Monitor
- RACF panels
- Enhancement to memory browse
- Extended highlighting
- 11 New SDSF primary panels
  - Not covered in this presentation
  - Numerous new columns on existing displays

SDSF Hidden Treasures

Rob Scott
Rocket Software
rscott@rs.com

Objectives

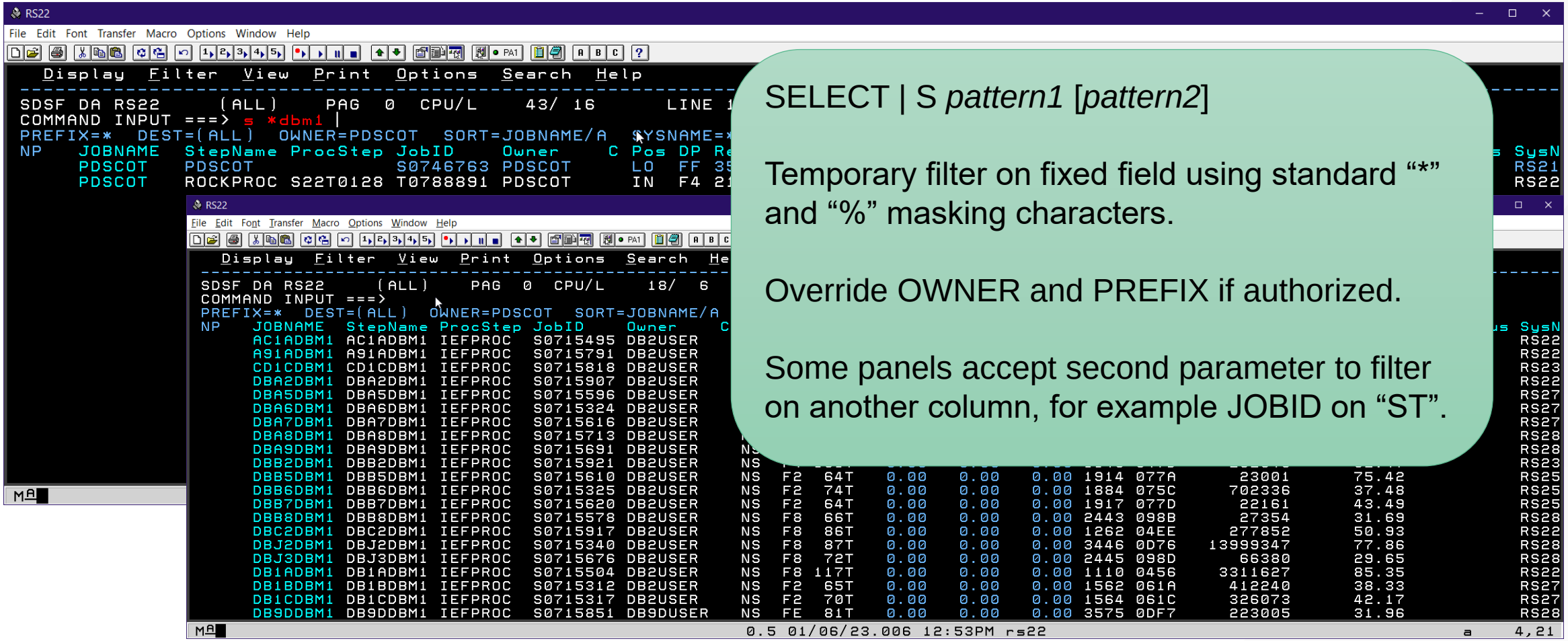
- Manipulating SDSF Tables
  - Set options
  - Fastpath Select
  - Locate Column
  - Arrange
  - Filter
  - Show
  - Block Actions
  - Overtype Extension
  - Snapshot
  - Point-and-shoot
- Browse Sessions
- Slash Extensions
- Searching Data Set Lists
- Running REXX inside SDSF

Manipulating SDSF Tables

SET Command

- SET command allows numerous options at the user level
  - Saved in ISPF profile
  - There are numerous other SET commands not described here
- SET DISPLAY ON | OFF
  - Show PREFIX, OWNER, DEST and SYSNAME above column titles
- SET ACTION ON | LONG | SHORT | OFF
  - Show valid action characters
- SET SCREEN
  - Change screen colours and highlighting
- SET CONSOLE *name*
  - Set EMCS console name for operator commands
- SET DATE MMDDYYYY | DDMMYYYY | YYYYMMDD *separator*
  - Set date format and optionally the separator character

Fastpath Select



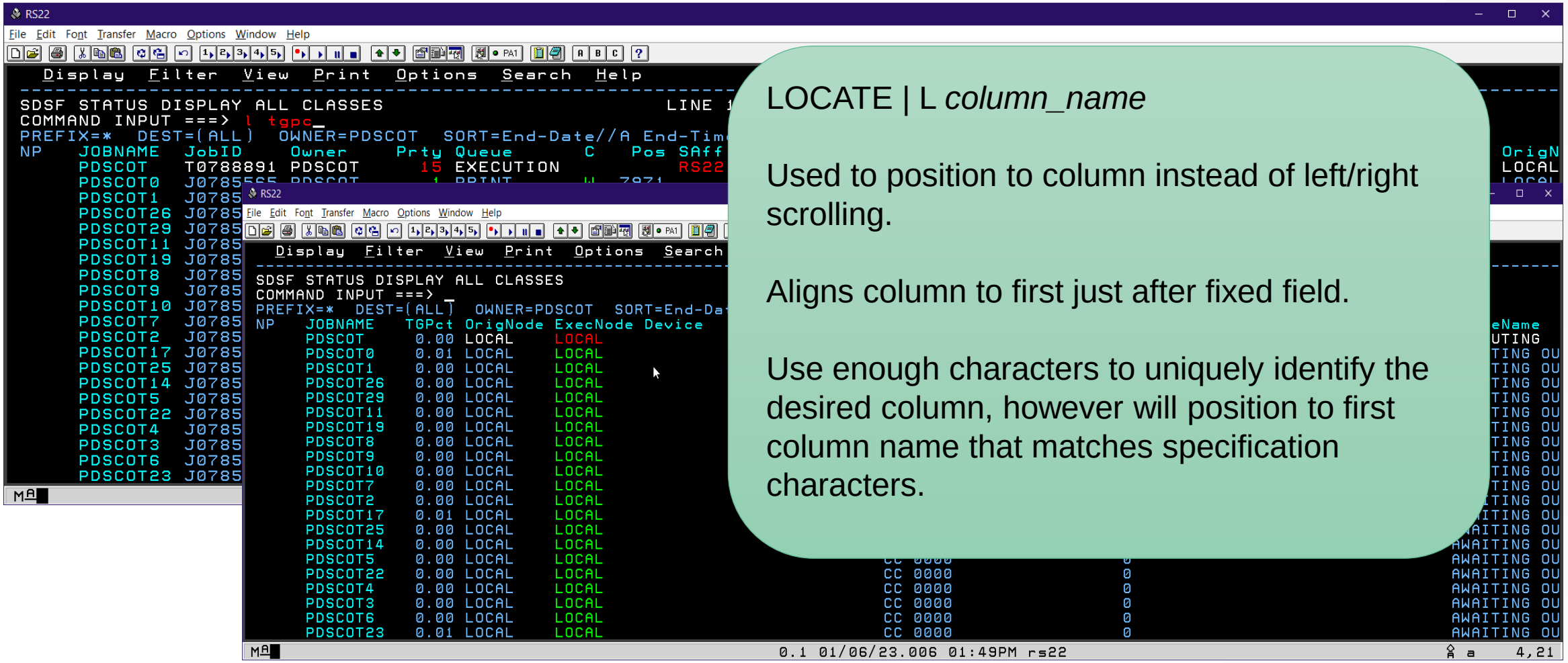
SELECT | S *pattern1* [*pattern2*]

Temporary filter on fixed field using standard “*” and “%” masking characters.

Override OWNER and PREFIX if authorized.

Some panels accept second parameter to filter on another column, for example JOBID on “ST”.

Locate Column



LOCATE | L *column_name*

Used to position to column instead of left/right scrolling.

Aligns column to first just after fixed field.

Use enough characters to uniquely identify the desired column, however will position to first column name that matches specification characters.

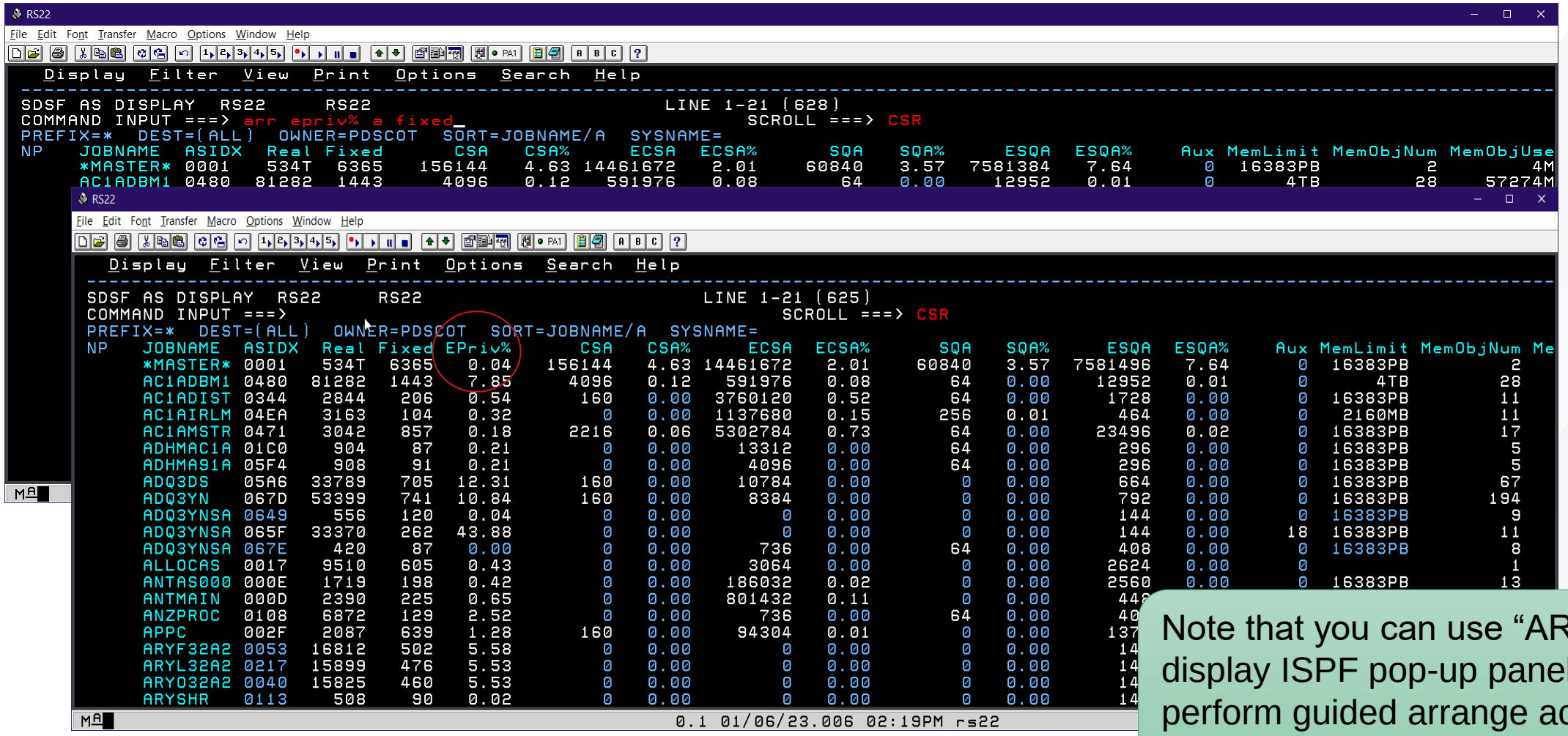
Arrange

ARR command to re-order columns on panel (save in ISPF profile)

Syntax

ARR *column_name* FIRST	Put column first
ARR *column_name* LAST	Put column last
ARR *column_name1* B *column_name2*	Put column before another
ARR *column_name1* A *column_name2*	Put column after another
ARR *column_name1* A .END	Hide column
ARR *column_name nn*	Set column width to nn
ARR DEFAULT	Reset column order

Arrange Column



Note that you can use “ARR ?” to display ISPF pop-up panel to perform guided arrange actions.

Filter

- Use FILTER | FIL command to select rows to be displayed
  - Applied after PREFIX, OWNER, DEST, and SYSNAME processing
  - Saved in ISPF profile
  - Use FILTER ? to drive pop-up while in ISPF to compose expressions with multiple filters

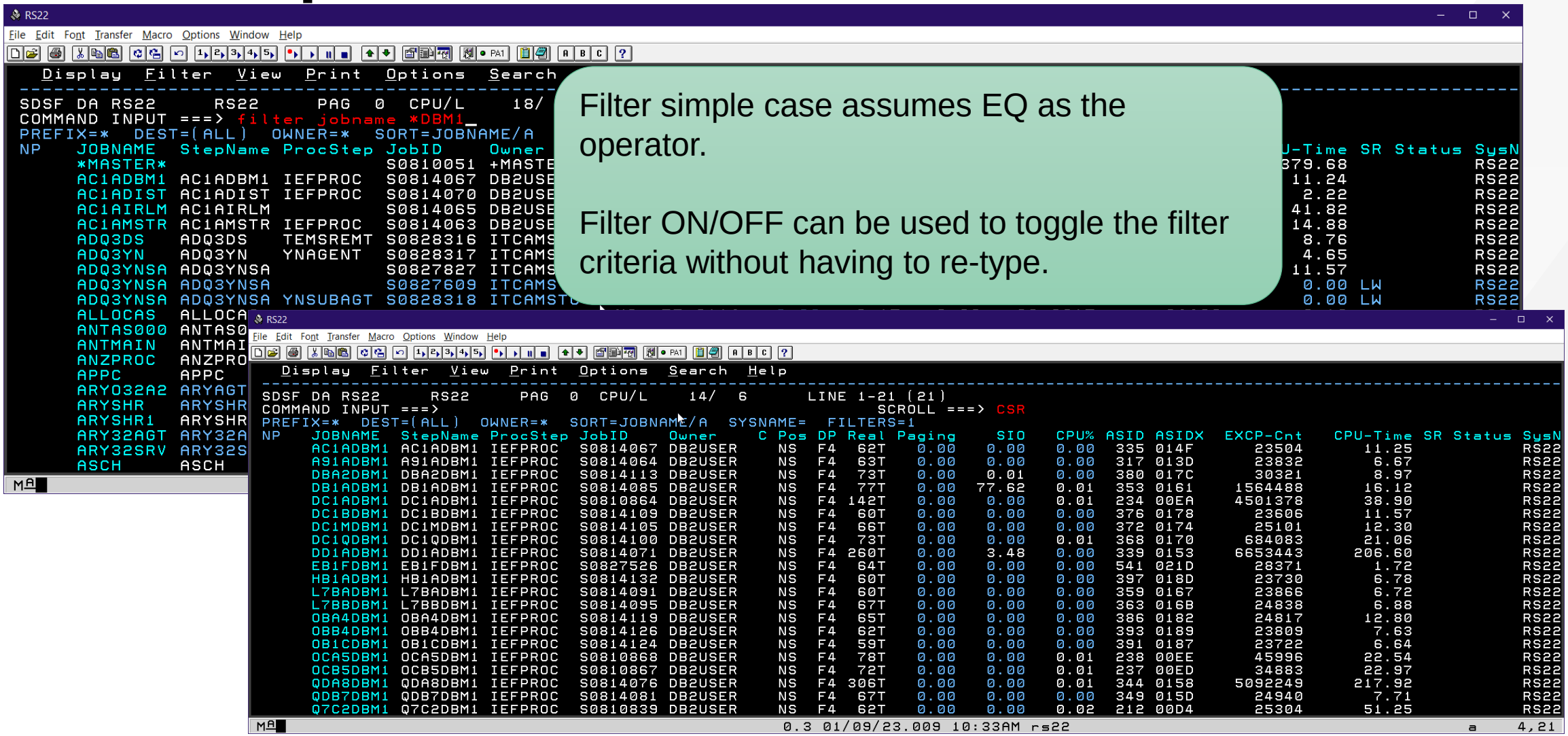
Syntax

FILTER ON | OFF	*Turn filtering on or off*
FILTER AND | OR	*How to treat multiple expressions*
FILTER *column_name [operator] value*	*Specify FILTER criteria*

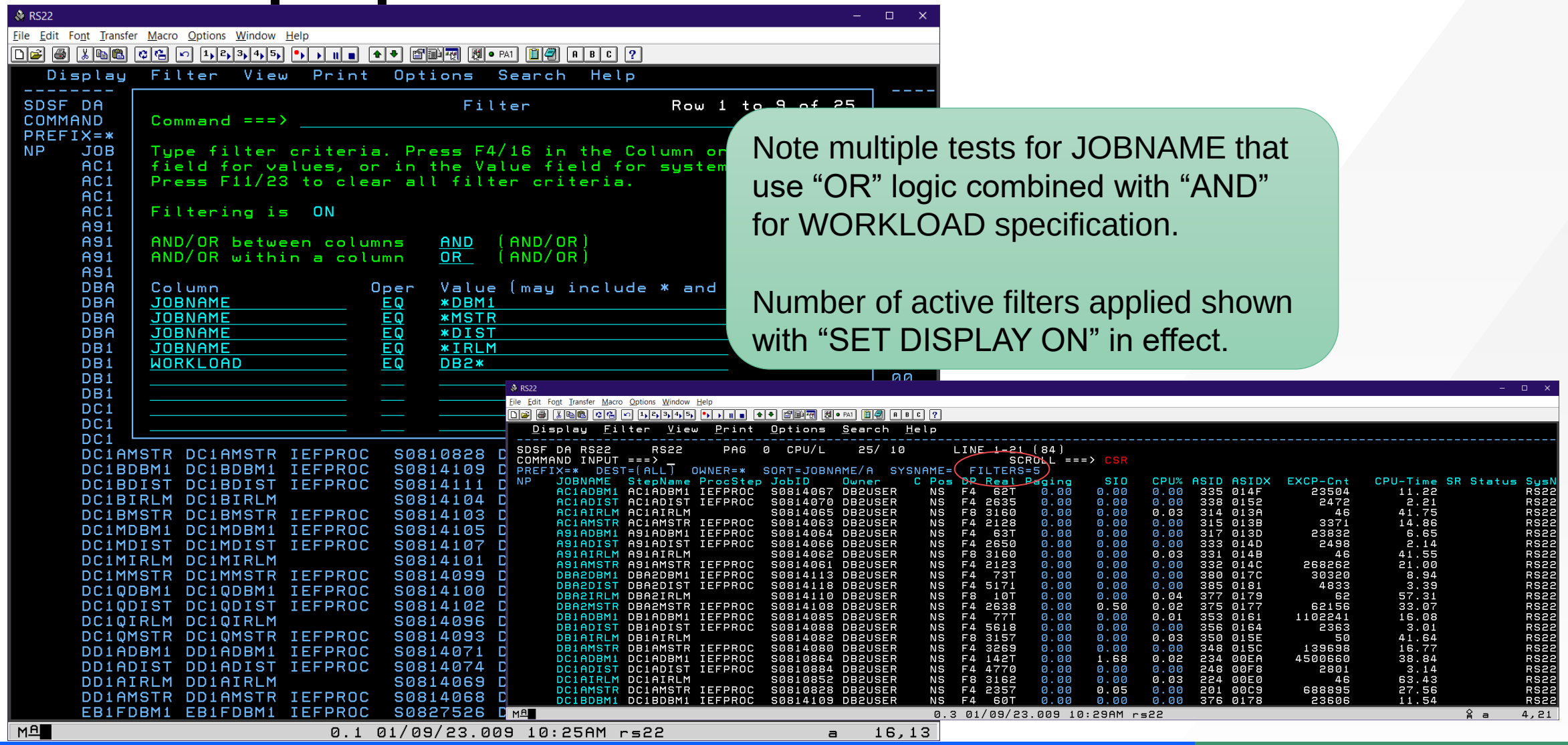
Operator values :

LT,LE,EQ,NE,GE,GT	Default operator is EQ
<,<=,=,<>,>,GE	

Filter Simple Case



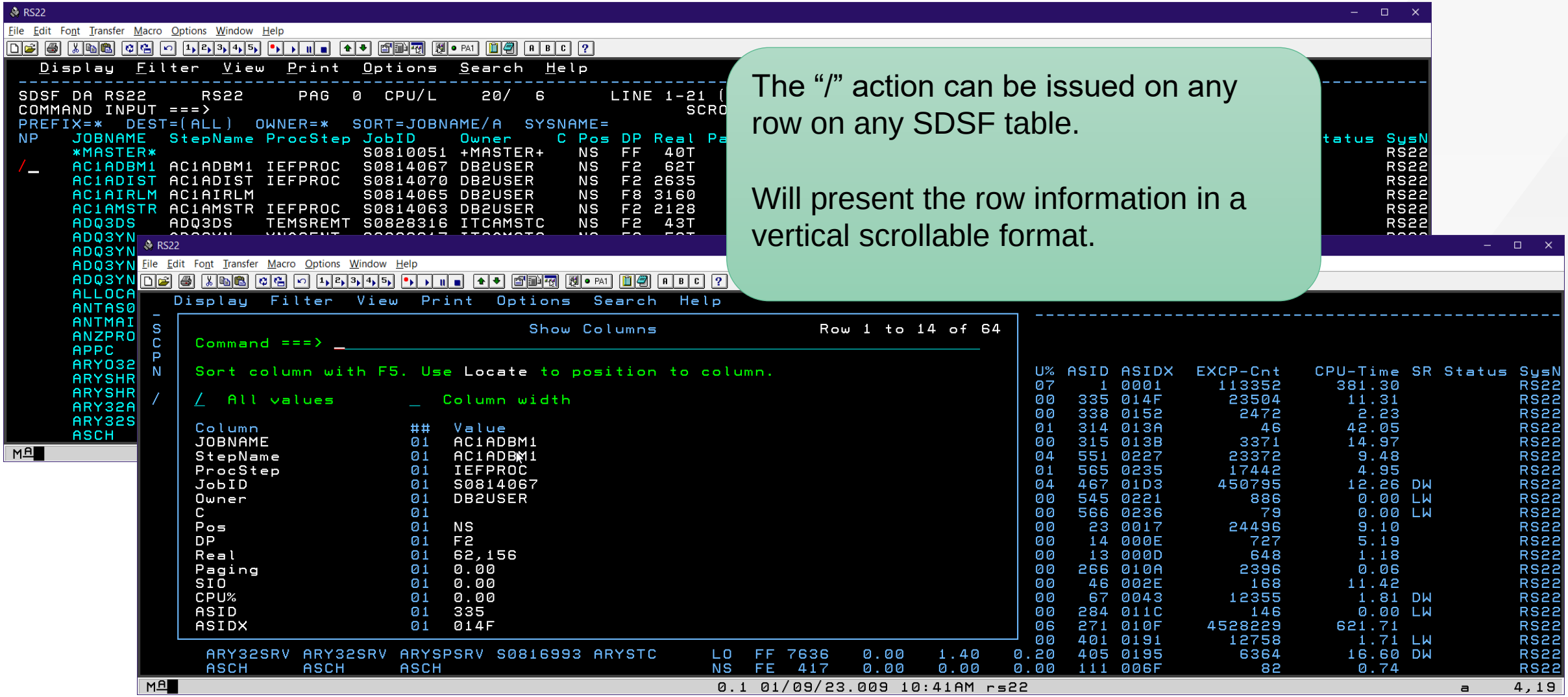
Filter Pop-Up



Note multiple tests for JOBNAME that use “OR” logic combined with “AND” for WORKLOAD specification.

Number of active filters applied shown with “SET DISPLAY ON” in effect.

Show Row Information

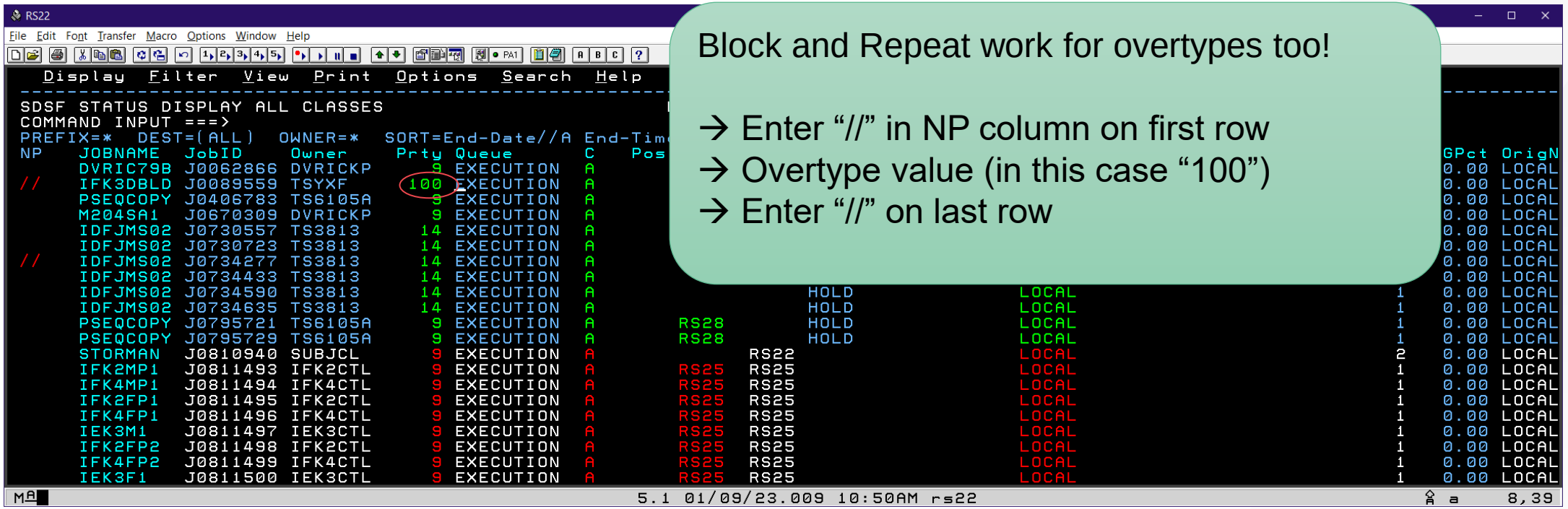


The “/” action can be issued on any row on any SDSF table.

Will present the row information in a vertical scrollable format.

Block and Repeat

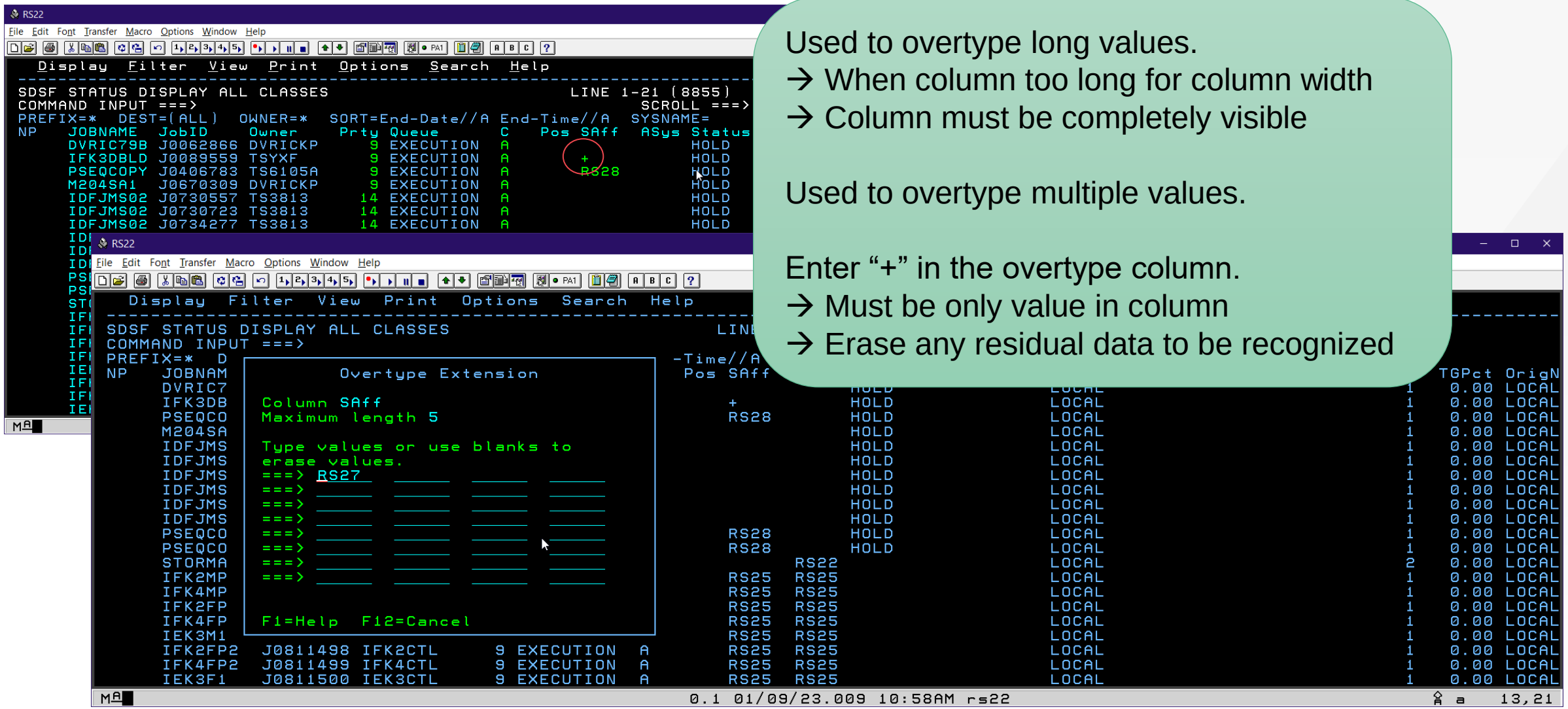
- Block command // applies an action against multiple rows
- Repeat command = repeats last action against the current row



Block and Repeat work for overtypes too!

→ Enter "//" in NP column on first row
→ Overtype value (in this case "100")
→ Enter "//" on last row

Overtype Extension

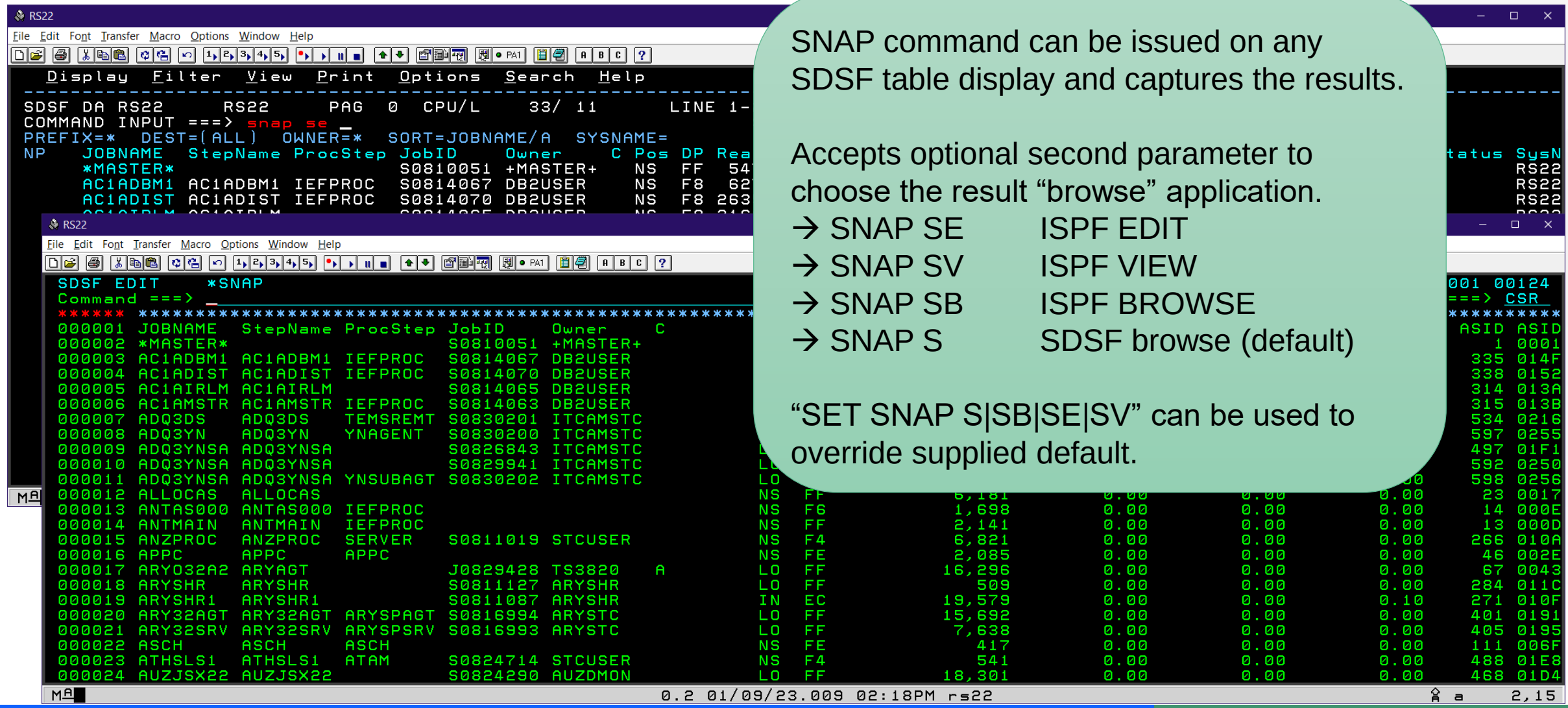


Used to overtype long values.
→ When column too long for column width
→ Column must be completely visible

Used to overtype multiple values.

Enter “+” in the overtype column.
→ Must be only value in column
→ Erase any residual data to be recognized

Snapshot



SNAP command can be issued on any SDSF table display and captures the results.

Accepts optional second parameter to choose the result “browse” application.

→ SNAP SE ISPF EDIT
→ SNAP SV ISPF VIEW
→ SNAP SB ISPF BROWSE
→ SNAP S SDSF browse (default)

“SET SNAP S|SB|SE|SV” can be used to override supplied default.

Point-and-shoot

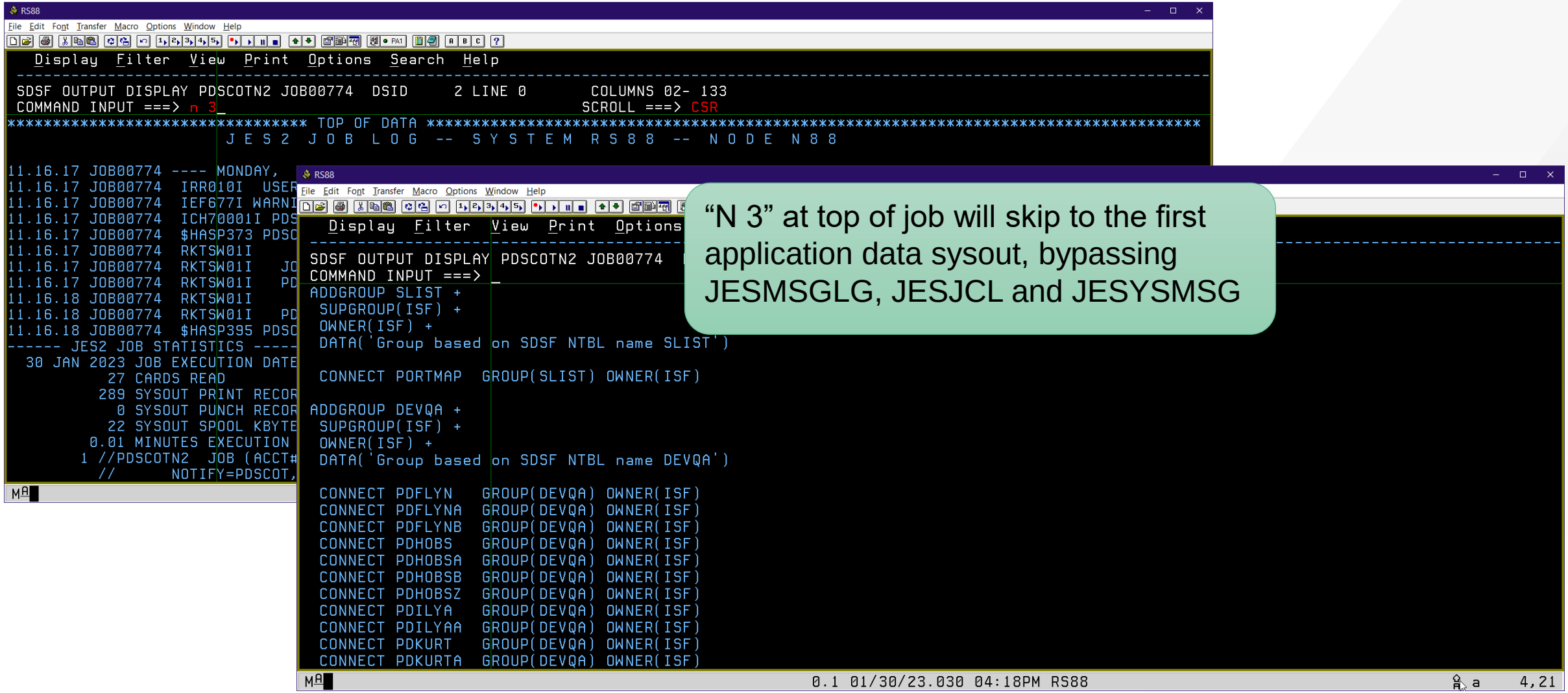
- SDSF supports various point-and-shoot functions
- NP
  - Invoke browse (or not) depending on SET BROWSE
- Fixed field
  - Invoke default action for the column
  - Example is JDS for JOBNAME on DA, H, I, ST and O panels
  - Can be toggled with SET PAS ON | OFF
    - Alias of SET FFPS
- Memory address fields
  - Invokes SDSF memory browse on the 31bit or 64bit address
  - Inherits the ASID and SYSNAME from the row (if present)

Browse Sessions

SDSF Browse

- Select action character for job output and other logical sources
  - S | SB | SE | SV Native SDSF browse
  - SB | SE | SV ISPF variants (Browse, Edit and View)
  - SET BROWSE S | SB | SE | SV | NONE
    - Point-and-shoot for NP – alternate to typing
- Following points apply to native SDSF browse only
- NEXT | N and PREV | P primary commands skip to next or previous logical dataset
  - Alternative to scrolling
  - Positions to start of data set
  - Option parameter of relative dataset number (default 1)
    - N 3 would skip three data sets
    - Can use S*n* as action character to browse and immediately skip *n* data sets

SDSF Browse



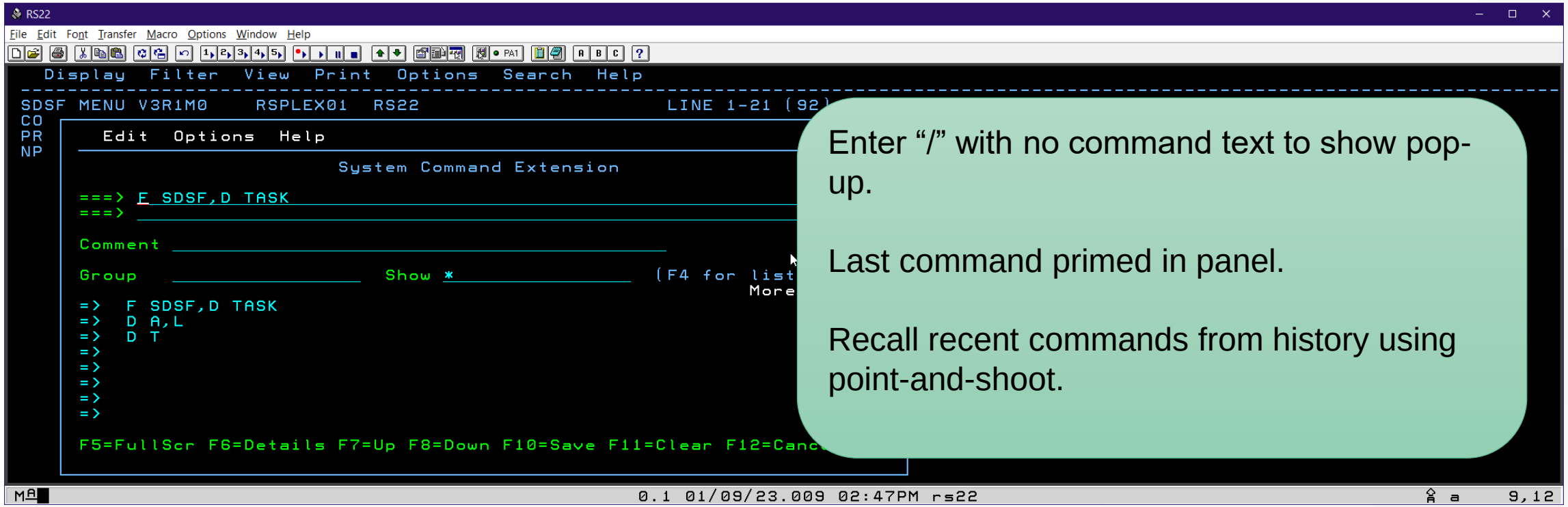
"N 3" at top of job will skip to the first application data sysout, bypassing JESMSGLG, JESJCL and JESYSMSG

Slash Extensions

Slash Command Tips

- Enter / with no text to drive pop-up with recently issued commands
- Use pop-up for mixed case command text
  - /cmdtext with always fold to uppercase
- Screen size greater than 36x143 enables single line entry for long operator commands (z/OS 2.5+)
  - Easier to manipulate long operator commands
- End your command with + to continue command using pop-up
  - Good for when running out of room on normal SDSF command input

Slash Command History

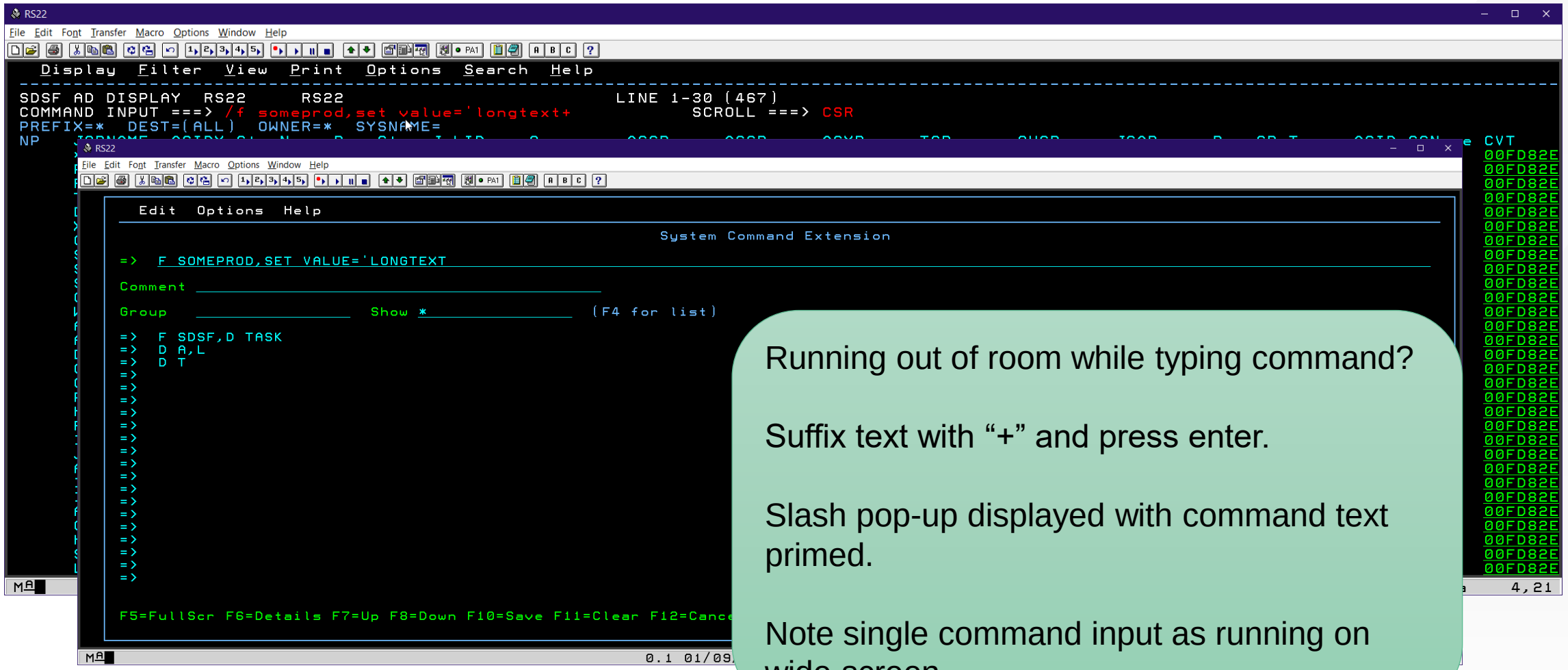


Enter “/” with no command text to show pop-up.

Last command primed in panel.

Recall recent commands from history using point-and-shoot.

Slash Command Extension



Running out of room while typing command?

Suffix text with “+” and press enter.

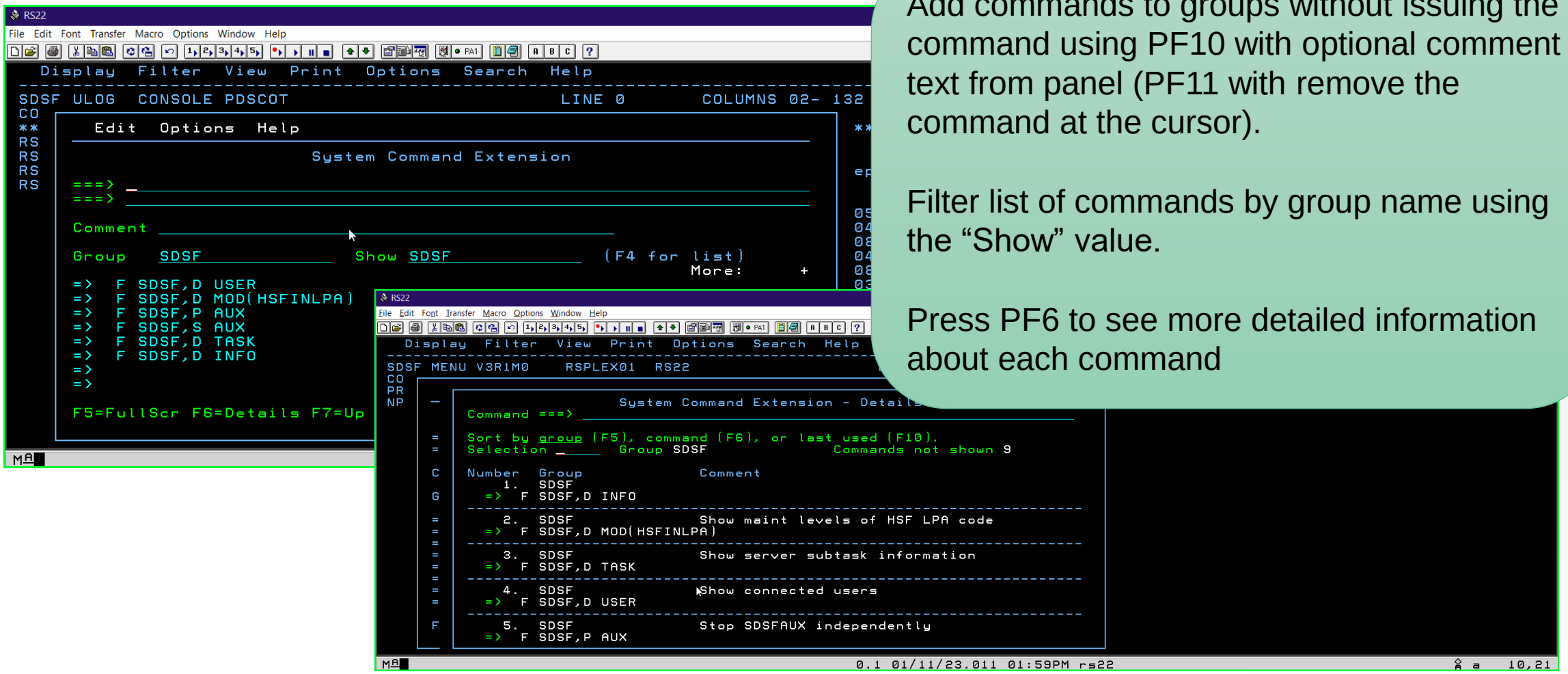
Slash pop-up displayed with command text primed.

Note single command input as running on wide screen.

Slash Command Groups

- Slash command pop-up allows you to :
  - Associate a command with a group
  - Associate a comment with a command
  - Display commands based on group pattern
  - List all command groups
  - Save a command without issuing it
  - Delete a single command, group of commands, or recent commands
  - View details about a command

Slash Command Groups



Add commands to groups without issuing the command using PF10 with optional comment text from panel (PF11 with remove the command at the cursor).

Filter list of commands by group name using the “Show” value.

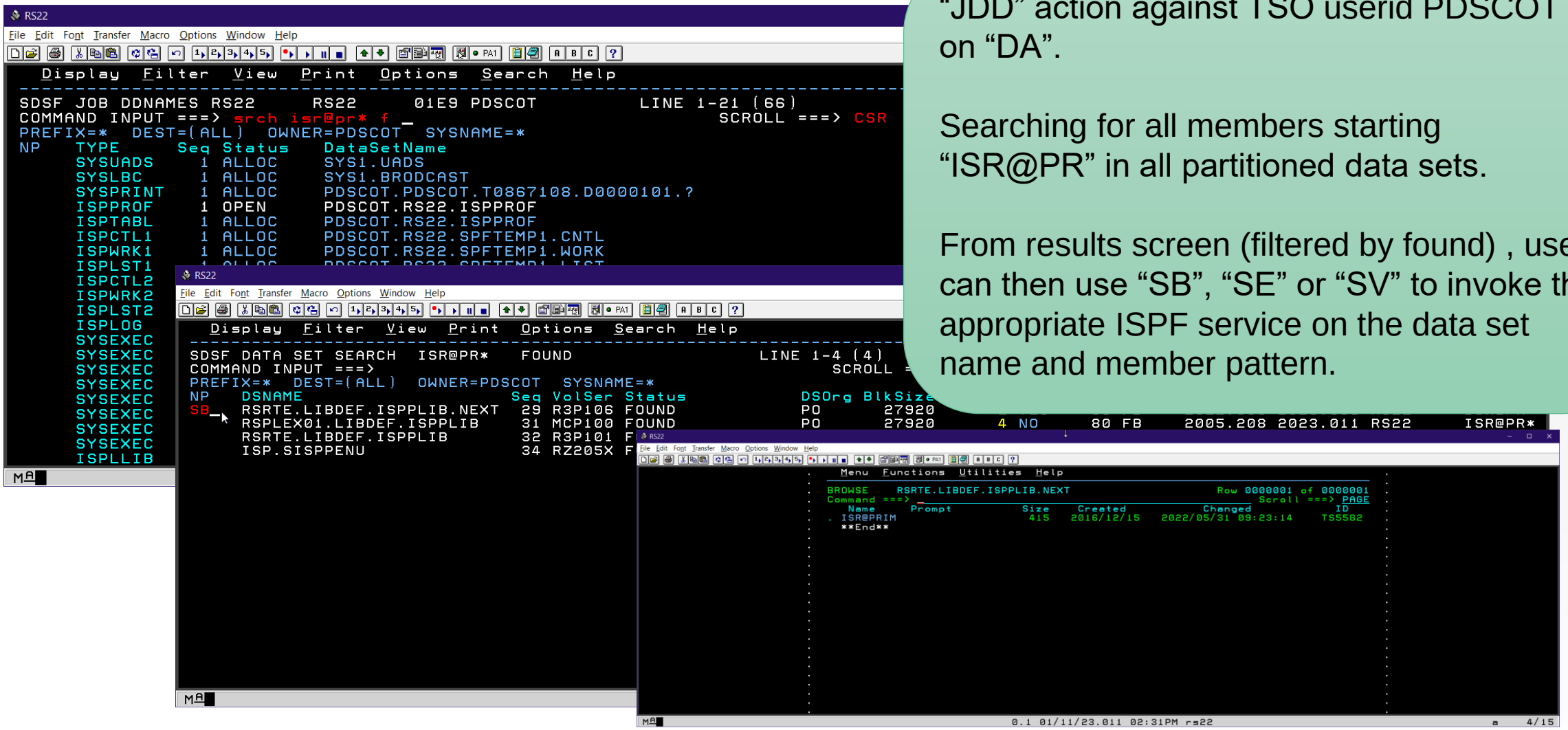
Press PF6 to see more detailed information about each command

Searching Data Set Lists

SDSF Displays With Data Set Lists

- Starting with z/OS 2.3, SDSF includes several displays that show lists of various types of data set
  - APF Authorized Load Libraries
  - LNK Link List
  - LPA Link Pack Area List
  - PARM Logical PARMLIB
  - PROC JES2 Proclibs
  - JDD Job Allocated Data Sets (action on DA and other panels)
- Use the SRCH command to search the list using a member name pattern
  - SRCH *member_pattern* [*F* | *NF* | *ALL*]
  - Member name pattern can include standard * and % masking characters
  - Optionally restrict results by :
    - Data sets containing member (F)
    - Data sets not containing member (NF)

Searching Data Set List



"JDD" action against TSO userid PDSCOT on "DA".

Searching for all members starting "ISR@PR" in all partitioned data sets.

From results screen (filtered by found) , user can then use "SB", "SE" or "SV" to invoke the appropriate ISPF service on the data set name and member pattern.

Running REXX Inside SDSF

Running REXX Against A Row

- Use % in NP column to drive pop-up panel that asks for REXX exec name and optional arguments
  - *%name* will immediately drive the "*name*" REXX exec (if found) bypassing the popup
  - You might need to increase NP width using the ARR command in order to have enough space to type *%name*
    - You can also type *+nn* in NP to set NP width to *nn* characters
- Exec passed arguments in following format :
  - *CurrentPanel PrimaryPanel RowToken PrimaryCmd* (*UserArgs*
  - RowToken required for any subsequent actions against the row
  - UserArgs contains any user specified arguments

Running REXX Against A Row

- Use “address SDSF ISFGET” to retrieve values for the row

```
address SDSF “ISFGET ‘”PrimaryCmd”’ TOKEN(‘”RowToken”’)”
```

- Use ISFQUERY() function to determine if EXEC was invoked under SDSF

```
rc=isfquery()
if rc<>0 then
do
  Say "** SDSF environment does not exist, exec ending."
  Exit 20
end
```

- Use ISFQUERY(“INIT”) to set environment variables

```
rc=isfquery("INIT")
Say "isfprefixwas set to:" isfprefix
Say "isfownerwas set to:" isfowner
Say "isfdestwas set to:" isfdest
```

-

Generating SDSF REXX With RGEN

- RGEN command generates starter exec you can modify
  - By default generates all commands to navigate to the panel you are on
  - RGEN X will display pop-up with list of examples
    - Select example and modify as required
  - Exec is generated to ISPF temporary data set
    - Be sure to copy to permanent data set prior to editing
  - Contains ISPF note lines for tips and tricks
    - Use ISPF MD line command to convert to data lines if required
- Help for SDSF REXX
  - REXXHELP | REXXH command inside product
  - COLSHELP | COLH command inside product
  - SDSF User Guide – Using SDSF With The REXX Programming Language
  - Share 24671 - Learn To Use SDSF Rexx