



Session 17578: ISPF Power Tools: Using Panel Exits to Dynamically Display/Hide Panel Lines

Pedro Vera
pedro@us.ibm.com
IBM Corporation



#SHAREorg

SHARE is an independent volunteer-run information technology association
that provides **education, professional networking and industry influence.**

Copyright (c) 2015 by SHARE Inc. Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/2.0/>



I am a developer in the DB2 Administration Tool.

I am a long time MVS system programmer

I am the UI champion for Information Management tools.

Today, I will talk about how to display or hide lines from a panel.

About Pedro



- Frequent contributor to social media
 - Ibmmainframes - TSO/ISPF forum (not part of IBM)
 - ISPF-L list
 - LinkedIn



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15

SHARE
in Orlando **2015**
2

I frequently use the Wimpy avatar, but that is my picture on Wimpy's employee badge.

Agenda

- Description of problem
- Demo
- Strategy to resolve
- How to implement
- Expand & Collapse example



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15

SHARE
in Orlando **2015**
3

The event logo for SHARE in Orlando 2015 features the text 'SHARE' in a large, bold, red font, followed by 'in Orlando' in a smaller black font and '2015' in a large, bold, red font. To the right of the text are three circular icons: a lightbulb, a globe, and a network diagram. Below the text and icons is a small number '3'.

Description



There is too much information on some panels.

- Some panels have several lines of information at the top of the panel.
- The information is useful at first, but is less useful after you become familiar with the product.
- If you can remove the information, more user data will be visible.
- This is a common problem and the solution has universal appeal.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE** in Orlando **2015**  4

Some panels have extra information at the top of the panel. The information tells the user how to use the panel.

The information is useful at first, but is less useful after you become familiar with the product.

If you can remove the information, more user data will be visible.

I have shown this to numerous people within IBM. The consensus is, is that this is a universal problem. I hope you find the solution useful.

DB2 Admin Tool: Typical Panels



- A list of primary commands are shown (line 1)
- Line commands along with a short description are shown (lines 2-6)
- The filter line allows the user to see a subset of the objects (line 7)

```
DB2 Admin                               DSNB Table Spaces                Row 1 to 11 of 72
Command ==> _____ Scroll ==> CSR_

1  Commands: GRANT MIG DIS STA STO ALL CT
2  Line commands:
3  T - Tables D - Database A - Auth G - Storage group ICS - Image copy status
4  DIS - Display table space STA - Start table space STO - Stop table space
5  ? - Show all line commands
6
7  Select Name      DB Name      Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
-----
_____ PSV1MXT1 DSN01373      1 BP0       R N A Y Y        1          -1      32 G Y
_____ PSV1MXT2 DSN01374      1 BP0       R N A Y Y        1          -1      32 G Y
_____ PSVT1    DSN02233      1 BP0       R N A Y Y        1          -1      32 G Y
_____ PSVTS302 DSN0PSV1     2 BP8K0     A N T N N        0          -1       4 G Y
_____ PSVTGH03 DSNDB04      1 BP0       A N A N N        1          -1      32 G Y
_____ PSVTS01  PSVDB01      1 BP0       A N A N Y        1          -1       4 G Y
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



8/17/15

5

The panels for the DB2 Administration Tool currently look like this figure. In the top half of the panel, we have:

- Line 1 is a list of primary commands. Those are commands that can be typed into the command line at the top of the panel. In Admin Tool, the user can also double-click to invoke the commands.
- Lines 2-6 have a list of Line commands (these are commands that are entered against an individual object in the list). The line command information at times takes up several lines of the screen.
- Line 7 is a filter line allows users to only see a subset of the objects from the list, based on the filter criteria specified.

The primary commands and line command sections serve as instructions on how to use the panel. Our design guidelines say that we should provide this information!

After the user becomes familiar with the product, some users want to hide some of these instructions. In this example, we can easily hide 6 lines. On a standard ISPF screen, that is 25% of the panel area.

Strategy



- Allow each individual user to customize the product.
- The user can show or hide sections of the panel.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE** in Orlando **2015** 

Our strategy to resolve this problem is to give individual users the ability to customize their panels through settings in the product options panel.

The user can toggle on or toggle off some sections of the panel.

This is a strategy that will be implemented across much of our Information Management tools portfolio.

The concept I am describing today is a DB2 Admin Tool implementation of that strategy.

This change is part of DB2 Admin Tool version 11.2

New Options Command



- 'OPTIONS' is available as a primary command.
 - The user can make changes at any point.
 - Select D to update display options
- The user can turn on and off sections of the panel.
- The user can select any combination of options.
- Press Enter to save and dismiss the panel.
- The settings are saved in the ISPF profile.

```
Enter "/" to select an option.  
_ Show the DB2 Admin action bar  
_ Show panel instructions  
/ Show primary commands  
/ Show line commands  
_ Hide descriptions  
/ Show the filter line
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15



There are new panel display options.

The Admin tool will support OPTIONS as a primary command. This will be a command available from any Admin Tool panel. The user can make changes at any point in the product.

The user can turn on and turn off sections of the panel by typing a slash character.

The user can select any combination of the display options.

Press the Enter key to save the settings and dismiss the panel. The options are saved in the ISPF profile. The next time you logon, the previous settings remain in effect.

Using the Panel Display Options



- The default is to behave as it has in the past. The user can select or deselect options as needed.
- These initial selections are made for you.

```
Enter "/" to select an option.  
_ Show the DB2 Admin action bar  
_ Show panel instructions  
/ Show primary commands  
/ Show line commands  
_ Hide descriptions  
/ Show the filter line
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE** in Orlando 2015  8

The product default will be to show the historic view of the panel. The panel will behave as it did in the past. The panels will show the primary commands, the line commands, and the filter line.

The user can select or deselect options as desired at any time after starting the Admin Tool.

Examples

- Some examples follow



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE** in Orlando **2015** 

Example 1: current view



- The default view will be to remain the same as before this feature is added.
- The user has to explicitly change the view.

```
Enter "/" to select an option.  
_ Show the DB2 Admin action bar  
_ Show panel instructions  
/ Show primary commands  
/ Show line commands  
_ Hide descriptions  
/ Show the filter line
```

```
DB2 Admin                               DSNB Table Spaces                               Row 1 to 11 of 72  
Command ==> _____                               Scroll ==> CSR_
```

Commands: GRANT MIG DIS STA STO ALL CT
Line commands:
T - Tables D - Database A - Auth G - Storage group ICS - Image copy status
DIS - Display table space STA - Start table space STO - Stop table space
? - Show all line commands

Select	Name	DB Name	Parts	Bpool	L	E	S	I	C	Tables	Act.	pages	Segsz	T	L
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
-----	PSV1MXT1	DSN01373	1	BP0	R	N	A	Y	Y	1	-1	32	G	Y	
-----	PSV1MXT2	DSN01374	1	BP0	R	N	A	Y	Y	1	-1	32	G	Y	
-----	PSVT1	DSN02233	1	BP0	R	N	A	Y	Y	1	-1	32	G	Y	
-----	PSVTS302	DSN0PSV1	2	BP8K0	A	N	T	N	N	0	-1	4	G	Y	
-----	PSVTGH03	DSNDB04	1	BP0	A	N	A	N	N	1	-1	32	G	Y	

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15

SHARE
in Orlando 2015



Example 1: This is the default view.

This is how the panel looks when only the primary commands, the line commands, and the filter lines are selected. This is the view that users are accustomed to seeing.

In the default view, it is showing rows “1 to 11” of the data.

Example 2: Primary Commands



- Show the primary commands.

```
Enter "/" to select an option.  
_ Show the DB2 Admin action bar  
_ Show panel instructions  
/ Show primary commands  
_ Show line commands  
_ Hide descriptions  
_ Show the filter line
```

```
DB2 Admin DSNB Table Spaces Row 1 to 17 of 72  
Command ==> Scroll ==> CSR
```

Commands:	GRANT	MIG	DIS	STA	STO	ALL	CT												
Select	Name	DB Name	Parts	Bpool	L	E	S	I	C	Tables	Act.	pages	Segsz	T	L				
_____	PSV1MXT1	DSN01373	1	BP0	R	N	A	Y	Y	1	-1	32	G	Y					
_____	PSV1MXT2	DSN01374	1	BP0	R	N	A	Y	Y	1	-1	32	G	Y					
_____	PSVT1	DSN02233	1	BP0	R	N	A	Y	Y	1	-1	32	G	Y					
_____	PSVTS302	DSN0PSV1	2	BP8K0	A	N	T	N	N	0	-1	4	G	Y					
_____	PSVTGH03	DSNDB04	1	BP0	A	N	A	N	N	1	-1	32	G	Y					
_____	PSVTS01	PSVDB01	1	BP0	A	N	A	N	Y	1	-1	4	G	Y					
_____	PSVTS60	PSVDB60	0	BP0	A	N	A	N	Y	1	-1	8	Y						
_____	PSVTS601	PSVDB601	0	BP0	A	N	A	N	Y	1	-1	4	Y						
_____	PSVTS604	PSVDB601	0	BP0	A	N	A	N	Y	2	-1	4	Y						
_____	PSVL2701	PSVDB701	0	BP8K0	L	N	A	N	Y	1	-1	0	0	Y					
_____	PSVLB701	PSVDB701	0	BP8K0	L	N	A	N	Y	1	-1	0	0	Y					

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Example 2 shows what the panel looks like when only the primary commands are selected by the user.

This is a much cleaner view. It is now showing rows “1 to 17” of the data.

End of Examples



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE** in Orlando **2015** 12

Summary



- 'OPTIONS' primary command is available from any panel.
 - The display options panel allows the user to select any combination of display options.
- The options can change the appearance of many panels.
 - Only panels that have been enabled are affected.
 - Scope: 100 panels
- It only takes three lines of code (sort of).

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE** in Orlando **2015** 13

The 'OPTIONS' primary command is available.

- The options panel allows the user to select any combination of display options.

The selections made in the options panel can change the appearance of many panels.

- However, this does not change the entire product.
- Only the panels that have been modified are affected.

To implement this is fairly simple. It only takes a few lines of code for every group of lines you want to hide.

I recommend that your implementation is similar so that our common users have a consistent experience.

How to Implement

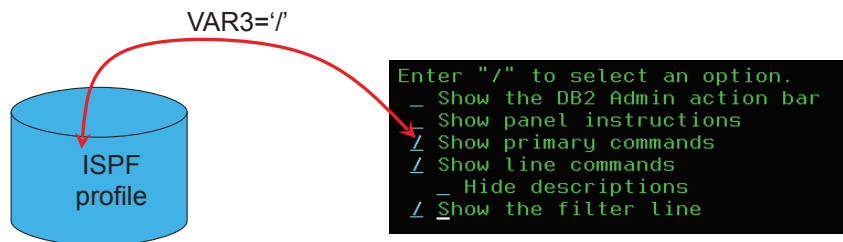
Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE** 
in Orlando **2015**
14

How Hiding is done:



- Define a panel to display options to the user.
- Save user settings to the ISPF profile.
 - You decide the variable name for each option.



Use VGET and VPUT as needed.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15

SHARE
in Orlando 2015
15

As a developer, you decide what variable names to use in your panels.

The entry fields on the options panel map to a variable name.

Save the user option variables to the ISPF profile so that the options are persistent across logons.

How it is done:



- Edit the panel source to specify the panel input exit.

```
) INEXIT LOAD, ISPPXMNX
```

```
. . .
```

```
<IF &VAR3= />
```

```
Commands: GRANT
```

```
</IF>
```

- <IF> statements control panel content.
- Exit described in Dialog Developer's Guide and Reference.
- Sample ISPPXMNX in ISP.SISPSAMP

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15



A feature in z/OS version 11 allows more control of panel contents. The name of the source input exit has to be specified in the panel. The first line of the panel definition has to be the)INEXIT tag. It provides the name of a load module. (first blue line in the example)

The exit determines if a line of the panel is displayed or not.

The exit understands a simple IF statement. Anything between a true <IF> statement and the ending </IF> is included in the panel. The example shows one nested line, but it could be any number of lines.

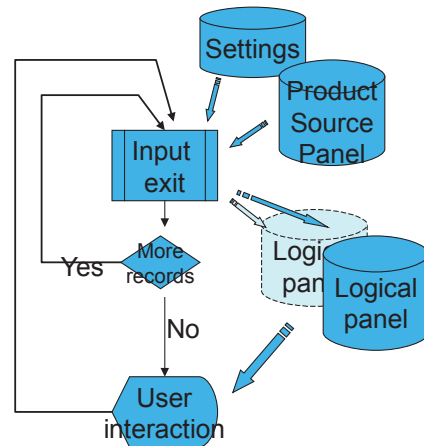
The slash, '/', is highlighted in red: It is the value of the variable specified in the OPTIONS panel. It is an unfortunate coincidence that a slash character is frequently used as an ending delimiter for tag languages. In this example, the slash is a constant.

Do not use any other embedded blanks in the <IF statement, except for the blank immediately after the <IF.

Everything should be in uppercase.

Overall logic

- The input exit gets called repeatedly, once for every source record of the panel.
- The exit sets a return code:
 - 0 – process the current record.
 - 2 – inserts a new record.
 - 4 – delete the current record.
 - 8 – stop calling the exit.



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando 2015
8/17/15 17

The exit is described in more detail in the Dialog Developer's Guide and Reference. There is a sample in SISPSAMP. Assemble and link the sample into your local loadmod library.

The panel input exit gets called as ISPF is reading the panel source. The exit gets called for every record of the panel.

The exit determines if a line of the panel will be displayed or not.

The exit sees each record of the panel source and when it finds any <IF> statement, it evaluates that line to determine whether to skip or include subsequent records.

The <IF> statements themselves are not propagated to the panel because return code 4 is set by the exit.

The exit sets a return code

- 0 – process the current record
- 2 – insert a record
- 4 – delete the current record
- 8 – stop calling this exit

Order of processing



1. Panel input exit, <IF> and all lines of the panel.
2. Execute the)INIT section
3. Display)BODY section
4. Execute the)PROC section

```
<IF &GRPCHK1=->  
.ATTR($V1)='CAPS (ON) '  
</IF>
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15

SHARE
in Orlando **2015**
18

This is the order of the processing:

1. Panel input exit – all lines of the panel
2. Then it executes the)INIT section
3. Then it displays the)BODY section
4. Finally, after an user response, it executes the)PROC section

All of the <IF> statements are processed before any of the normal panel logic is performed.

Some people are confused because you see the lines right next to each other. They are not executed at the same time, however.

Rules for Writing <IF . . . >



- “<” in column 1
- IF must be in uppercase
- The only space is after the <IF
- Only tests for equal condition
- Case must be exact
- End with “>”
- Example:

```
<IF &USRTASK=UNIX>  
.  
.  
.  
</IF>
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE** in Orlando **2015** 19

Here are the rules for writing the <IF> statements:

- The “<” sign must start in column 1
- The IF must be in uppercase characters
- There is only one space and it is after the <IF
- The variable name is on the left.
- Use an ampersand.
- It only tests for equal condition
- The case of the constant must be exact
- The statement ends with “>”

For Example:

```
<IF &USRTASK=UNIX>  
.  
.  
.  
</IF>
```

Expand and Collapse Example

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE** in Orlando **2015**  20

Expand / Collapse Example



- You can collapse groups by double clicking on the section heading.
 - '-' means collapse is available
 - '+' means expand is available

```
Change Session Profile
Command ==> _____
Type information, then press Enter.
- Security Control
  PC password . . . . . _____
  z/OS password . . . . . _____
- Printers
  PC printer port . . . . . _____
  z/OS printer name . . . . . _____
```

```
Change Session Profile
Command ==> _____
Type information, then press Enter.
+ Security Control
+ Printers
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

I have another example. This is a more simple example and one that you can probably do right away.

In this sample panel, there are several input fields, grouped by a category.

You can collapse the groups by double clicking on the group heading.

On the far left of the group heading:

- '-' means collapse is available (this means it has already been expanded.)
- '+' means expand is available (this means it has already been collapsed.)

Lets walk through the code that allows this function.

Driving Exec



- This rexx program displays a panel until F3 is pressed.

```
/* rexx */
grpchk1='- ' /*initial setting*/
grpchk2='- ' /*initial setting*/
Address ISPEXEC
rc = 0
"addpop"
Do while (rc =0)
  "DISPLAY PANEL(fig56) "
End
"rempop"
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE** in Orlando 2015 22

This is a rexx program that keeps displaying a panel until F3 is pressed.

Notice that there are two variables, GRPCHK1 and GRPCHK2, that are initialized. They control the initial view of the groups of input fields.

Panel Definition, part 1



```
) INEXIT LOAD, ISPPXMNX
) ATTR DEFAULT (%+_)
% TYPE (TEXT)    color (BLUE)
+ TYPE (TEXT)    color (green)
_ TYPE (INPUT)   color (turq) hilite (uscore)
? TYPE (OUTPUT) color (BLUE) CAPS (OFF)
\ TYPE (OUTPUT) color (turq)
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE** in Orlando 2015 23

I am showing parts of the panel so that I can point things out and it is easier for you to see. I will show it in three parts.

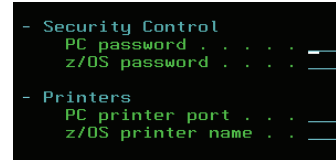
This is the top section of the panel. The very first line identifies the name of the panel exit.

It is fairly standard after that.

Panel Definition, part 2



```
)BODY expand(!!) width(45)
%! ! Change Session Profile ! !
+Command ==>_zcmd ! !+
+
\z?grptxt1
<IF &GRPCHK1=->
  + PC password . . . . ._ $V1      +
  + z/OS password . . . . ._ $V2    +
</IF>
+
\z?grptxt2
<IF &GRPCHK2=->
  + PC printer port . . . ._ $V4    +
  + z/OS printer name . . ._ $V5   +
</IF>
+
```



There are two Z variables:
GRPCHK1 and GRPCHK2

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15



This is the main body of the panel.

Things to note here are the <IF> statements that are before each group of fields. What it says is: if the variable contains a minus sign, include these lines in the panel.

Also, there is a closing </IF> statement.

Also notice that there are two Z variables: GRPCHK1 and GRPCHK2.

Each of the Z variables is referred to in the <IF> statements.

Panel Definition, part 3



```
)INIT
&grptxt1 = 'Security Control'
&grptxt2 = 'Printers'
.zvars   = '(GRPCHK1 GRPCHK2 )'
```

```
- Security Control
  PC password . . . . .
  z/OS password . . . . .

- Printers
  PC printer port . . . . .
  z/OS printer name . . . . .
```

```
)PROC
/* if the cursor is on group heading, flip status*/
If (.cursor = 'GRPTXT1','GRPCHK1')
  If (&GRPCHK1 = '-') &GRPCHK1 = '+'
  Else                  &GRPCHK1 = '-'

If (.cursor = 'GRPTXT2','GRPCHK2')
  If (&GRPCHK2 = '-') &GRPCHK2 = '+'
  Else                  &GRPCHK2 = '-'

)END
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15



This is the)PROC section of the panel.

It uses the .CURSOR function to determine the location of the cursor. If the cursor is on the group heading, it flips the value of the expand/collapse indicator. It flips from '+' sign to '-' sign.

The next time the panel is displayed, the <IF> statement is resolved differently and the content of the panel is different. The content is either hidden or shown.

Remember that the rexx program keeps displaying the panel until PF3 is pressed.

Implementation in DB2 Admin Tool



The code is similar to the expand / collapse example except:

- The controls are in a separate popup panel.
- The implementation was more complicated:
 - We added an action bar at the same time.
 - Implemented in over 100 panels.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE**
in Orlando 2015 26

The code used for the DB2 Admin Tool is similar to the expand / collapse example except:

- The controls are in a separate popup panel.
- The actual implementation was more complicated because:
 - ✓ We added an action bar at the same time.
 - ✓ And because we implemented in over 100 panels.
 - ✓ Also, we used a modified panel exit to do other internal functions.

Gotcha!



Some functions operate on variables that are included in the)BODY section.

When you hide parts of the panel body:

- The .ZVARS variable may be affected
- .ATTR() has to be avoided for those collapsed fields
- Avoid REFRESH() function for those collapsed fields
- Some examples:

```
.zvars = '( + <IF &GRPCHK1=->  
<IF &GRPCHK1=+> .ATTR($V1)='CAPS(OFF)'  
$V1 + </IF>  
</IF>  
)'
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15

SHARE
in Orlando **2015**
27

There are some things to watch out for:

There are some ISPF functions that operate on the fields that are included in the body of the panel.

When you hide parts of the panel body, those functions will get a “DIALOG ERROR”:

- The .ZVARS variable may be affected. Do not specify fields that are **not** included in the)BODY.
- The .ATTR() function has to be avoided for those collapsed fields.
- Likewise, avoid the REFRESH() function for those collapsed fields.

Gotcha!



ISPF loads the panel exit into storage.

If the exit is from ISPLLIB, it loads it and will not load it again until you exit from ISPF.

If it is LIBDEF'ed, it seems to work okay.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE**
in Orlando 2015 28

The panel exit is a load module. ISPF loads it into storage when panel first references it.

If the exit is from the ISPLLIB concatenation, it loads it and will not load it again until you restart ISPF. My implementation was actually a modified version of the exit. This caused some confusion at first because as we made changes, the old behaviour continued.

My observation is that if it is LIBDEF'ed, it seems to work okay.

Debugging



Command ==> ISPDPTRC READ (DETAIL)

TLD#	Type	Panel	Section Cd	RC	Data
TLD1	Read	IBM21	-----		DD=ISP10609+3 DSN=IBM.SIBMPLIB
TLD1	Read	IBM21)INEXIT LOAD,ISPPXMNX
TLD1	Read	IBM21)attr format(mix)
TLD1	InEx	IBM21		0)attr format(mix)
.					
TLD1	Read	IBM21			<IF &SYSNODE=MYMVS1>
TLD1	InEx	IBM21		4	
TLD1	Read	IBM21			Group name . ._GROUP +
TLD1	InEx	IBM21		4	
TLD1	Read	IBM21			</IF>
TLD1	InEx	IBM21		4	
TLD1	Read	IBM21			+
TLD1	InEx	IBM21		0	+

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE** in Orlando 2015 29

Using the panel input exit adds another level of complexity for problem determination.

The panel input exit composes the contents of the panel source dynamically.

You can use ISPDPTRC with the READ(DETAIL) parameter to add the results from the panel input exit to the panel trace.

This is an example of what you might see.

RC = 0 means that the record is included in the panel

RC = 4 means that the record is excluded

<IF> statements are excluded.

You might recall from an earlier slide, the exit can modify the line. Record type= 'InEx' shows you the contents of that line.

Installation of the Panel Input Exit



- The exit source is in samplib.
- Assemble and linkedit.
- The library should be concatenated to ISPLLIB
 - You can LIBDEF to ISPLLIB
 - Or any data set in search order.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE** in Orlando **2015**  30

While the source for the exit is provided, you need to make it available for execution.

- The exit is in samplib.
- You need to assemble and linkedit.
- The library should be concatenated to ISPLLIB
 - ✓ You can LIBDEF to ISPLLIB
 - ✓ Or any data set in search order for load modules.

RFE



- There are two open requirements that facilitate the use of the input exit
 - 54810 provide load module
 - 54812 support input exit in DTL
- Please visit IBM Developerworks and vote for these RFE's.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE** in Orlando **2015**  31

There are two open requirements that facilitate the use of the input exit

- 54810 provide load module
- 54812 support input exit in DTL

Please visit IBM Developerworks site and vote for these RFE's. You need to register a userid in order to vote.

While I work for IBM, but they try to answer customer requirements first.

RFE 54810



- Requirement 54810 - Provide a load module
- ISPF provides the source code for exit ISPPXMNX in the samplib library. The function provided by this exit is very useful. It would be helpful to users of the exit if IBM delivered a compiled version of the exit.

https://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&CR_ID=54810

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

8/17/15 **SHARE**
in Orlando 2015 
32

Requirement 54810

- ISPF provides the source code for exit ISPPXMNX in the samplib library. The function provided by this exit is very useful. It would be helpful to users of the exit if IBM delivered a compiled version of the exit.

RFE 54812



- Requirement 54812 - Support)INEXT in DTL
- ISPF panels support the use of an)INEXIT statement, which specifies the name of an exit.
- Dialog Tag Language (DTL) is a tag-based language used to define panels. The Dialog Tag Language alleviates the developer from tedious specification of colors and from tedious alignment of text. However, DTL does not provide a tag statement that will produce the)INEXIT statement.

https://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&CR_ID=54812

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando **2015**
8/17/15 33

Requirement 54812

- ISPF panels support the use of an)INEXIT statement, which specifies the name of an exit.
- Dialog Tag Language (DTL) is a tag-based language used to define panels. The Dialog Tag Language alleviates the developer from tedious specification of colors and from tedious alignment of text. However, DTL does not provide a tag statement that will produce the)INEXIT statement.