ISPF IS DEAD! LONG LIVE ISPF!

User-Driven Enhancements Such as PDSEGEN and the CBT Usermod Collection for ISPF (CUCI)

Session 24946

Wednesday, August 17, 2019 3:30 PM

Thomas Conley
Pinnacle Consulting Group, Inc. (PCG)
pinncons@rochester.rr.com

Abstract

Have you heard that ISPF is dead? Do you still rely on ISPF to do your job? Do you want to continue to see enhancements to ISPF so it remains relevant to the work you do? If you answered "YES" to any of these questions, then this session is for you! To paraphrase Mark Twain, reports of ISPF's death were an exaggeration. A dedicated core of long-time ISPF users and developers are working to enhance ISPF wherever possible, in order to ensure that ISPF remains viable well into the future. Two significant works from this group have been released so far: PDSEGEN and the CBT Usermod Collection for ISPF (CUCI). Attend this session to learn about exploiting PDSE Member Generations, EDIT highlighting for unsupported languages such as Java, SQL, shell script (and many more!), and other useful ISPF usermods as well. We fully expect this effort to continue and grow into the future, so please, COME JOIN THE REVOLUTION!

Agenda

- ISPF is Dead...
- LONG LIVE ISPF!
- The ISPF Cabal
- PDSEGEN
- CBT Usermod Collection for ISPF (CUCI)
- REVEDIT
- PDS86
- Finally...

ISPF is Dead...

- IBM's investment in ISPF has been steadily dwindling
- z/OS V2R1 had 5 pages of enhancements
- z/OS V2R2 had 3 pages of enhancements
- z/OS V2R3 had two line items
  - TSO 8-character support
  - Make enhanced statistics usable (really a bug fix for an unusable design)
- z/OS V2R4 Preview offered these "enhancements"
  - ISPF Upper Case function for Japanese clients (available for V2R3)
  - Withdrawal of ISPF Workstation Agent (WSA) after V2R4

ISPF is Dead...

- ISPF is no longer strategic for IBM
- z/OSMF strategic platform going forward for sysprogs
  - Workflows now strategic for software installs, so get with the program
- IBM Developer for z Systems strategic platform for developers
  - Support to bring your own IDE (Eclipse, vscode)
- Future enhancements likely limited to infrastructure support, BUT...
  - PDSE Member Generations cannot be displayed on a member list
  - You can Edit/View/Browse member generations by guessing
  - But how do you know if they’re even there?
  - ISPF was not updated to support extended GDG's with > 255 generations
  - ISPF is not even being updated to support newer operating system options

LONG LIVE ISPF!

- Enter The ISPF Cabal
  - Tom Conley, Pinnacle
  - Dan Dalby, Compuware
  - Lionel Dyck, 21st Century
  - Sam Golob
  - John Kalinich
  - Bruce Koss, Wells Fargo
  - Ray Mullins, Trident Services
  - Greg Price, HCL
  - Bill Smith, Trident Services

The ISPF Cabal

- Dedicated team of ISPF users determined to keep ISPF relevant
- We feel z/OSMF adoption as new sysprog interface has been slow
- Many sites still use critical homegrown ISPF applications in-house
- ISPF still has a large, dedicated user base
- To support them, we will deliver function to enhance ISPF operation
- We feel ISPF's lifespan is easily another 15-20 years, or more
- We're dedicated to keeping ISPF relevant during that timeframe

PDSEGEN

- IBM introduced PDSE Member Generations in December 2013
- Unfortunately, it was rushed to GA
- Still today, no way to list PDSE Member Generations natively in z/OS
- ISPF Edit/View/Browse interface very limited and hard to use
- No support in IEBCOPY, JCL, DYNALLOC, et al.
- Lionel Dyck created PDSEGEN to address these shortcomings
- Other products that support PDSE Member Generations
  - CA-PDSMAN
  - Compuware FileAid
  - IBM Dataset Commander
  - MacKinney SimpList

PDSEGEN

- In May of 2016, Lionel began working on PDSEGEN
- Used PDSE Developer Tom Reed's presentation from SHARE in Seattle
- Presentation provided Rexx samples to access member generations
- Lionel expanded Rexx code to start working with member generations
- John Kalinich discovered Xephon code to read PDSE object libraries
- Greg Price modified code to read PDSE data libraries
- Incorporated into PDSEGEN V2.0 in June 2016
- If you want to use PDSE Member Generations, you need PDSEGEN
- ISPF unlikely to ever duplicate function available in PDSEGEN
- PDSEGEN available here and at Lionel Dyck's website

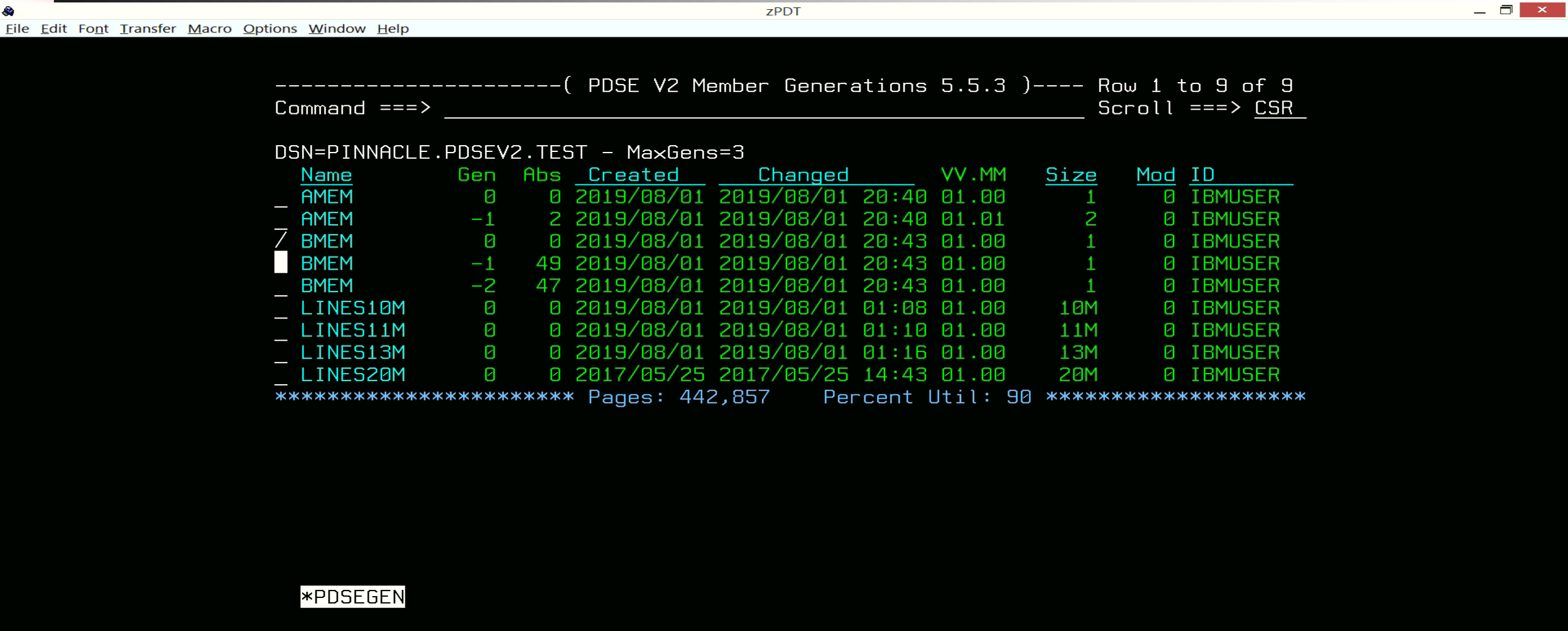
PDSEGEN and PDSE APARs

- PDSEGEN is great example of user-driven code for ISPF
- PDSEGEN development surfaced many PDSE Member Generation bugs
- DFSMS APAR OA42358, ISPF APARs OA42247, OA42248 added PDSE member generation support
- To safely use PDSE Member Generations, apply these APARs:
  - OA57897 - Oldest member gens inaccessible using ISPF Edit/Browse/View
  - OA56897 - AE OA56619 FIX COMPLETION Missing aliases after pending delete
  - OA56619 - Missing aliases after pending delete
  - OA55672 - PDSE member create failed with ABEND002 RC94
  - **OA54890 - IEBPDSE detects corruption after editing non-0 generation**
  - OA54281 - STOW RECOVERG bad RC for program object member gen

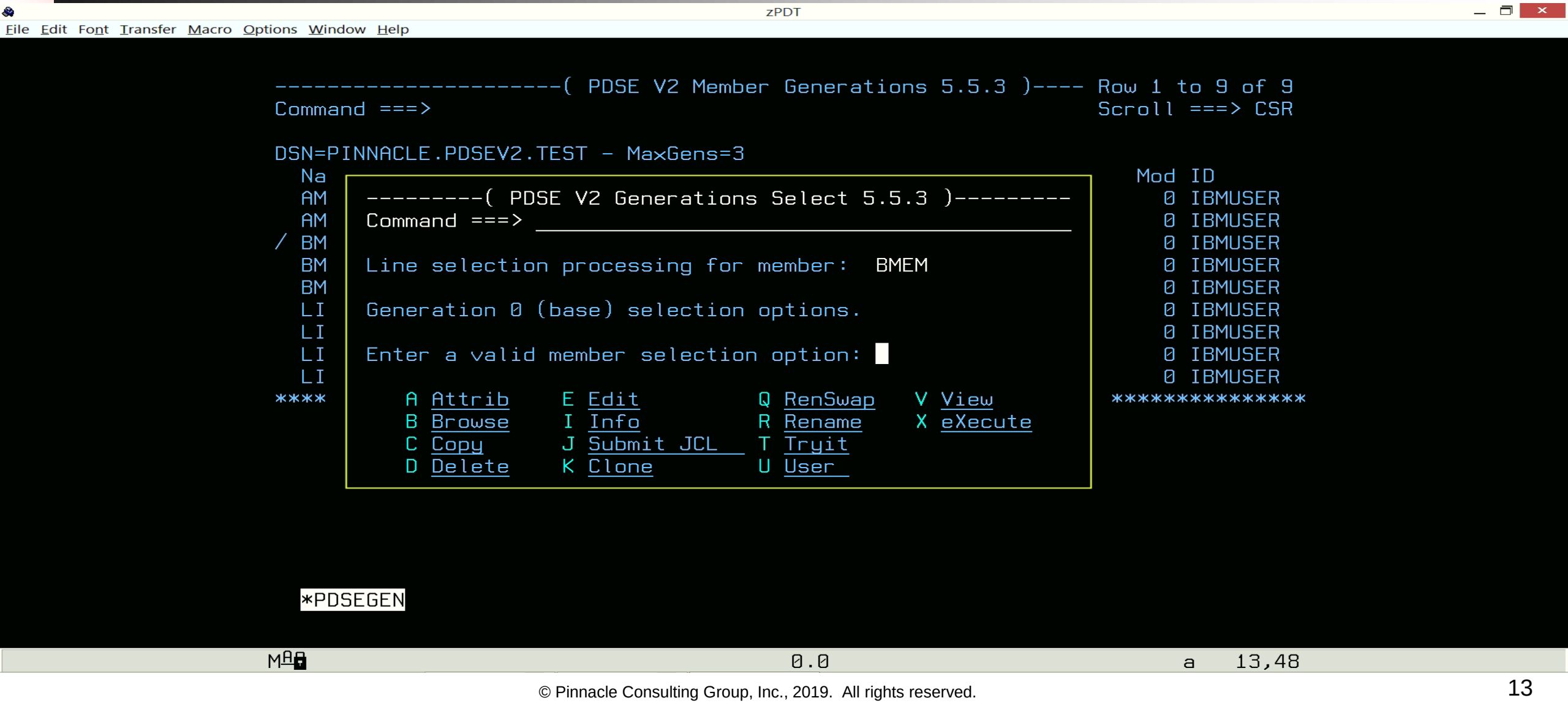
PDSEGEN and PDSE APARs

- To safely use PDSE Member Generations, apply these APARs (cont'd):
  - OA53942 - ABEND 0C1 occurred while writing to a PDSE
  - OA53621 - Error updating member generation directory STOW TYPE=RG
  - OA52543 - Using DESERV FUNC=GET_G and GET_ALL_G invalid length
  - OA51579 - IEC911I 315 ABEND when STOWing member at MAXGENS
  - OA51441 - Current/highest gen number not available to ISPF edit macros
  - OA51009 - Error msg for SAVE NEWGEN/NOGEN if gens not enabled

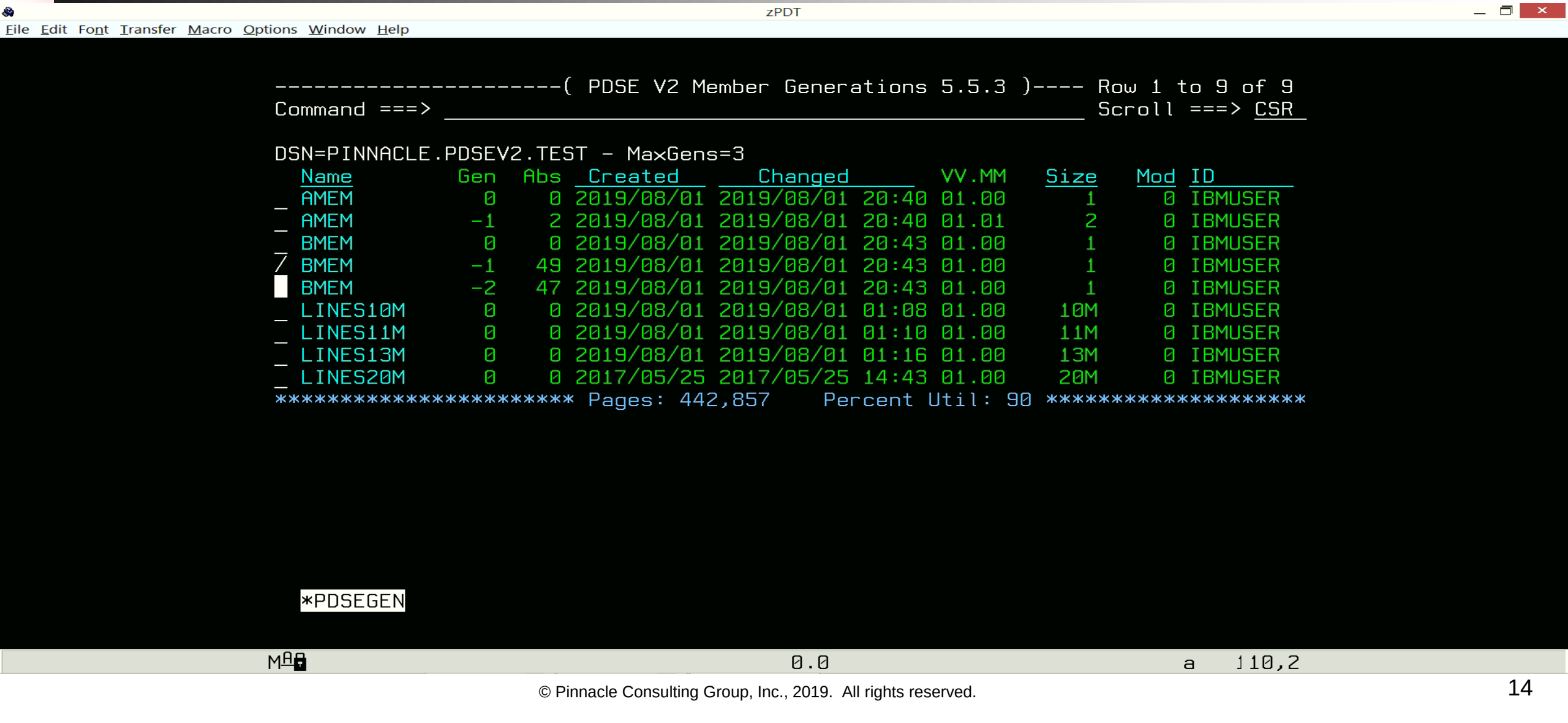
PDSEGEN Line Commands



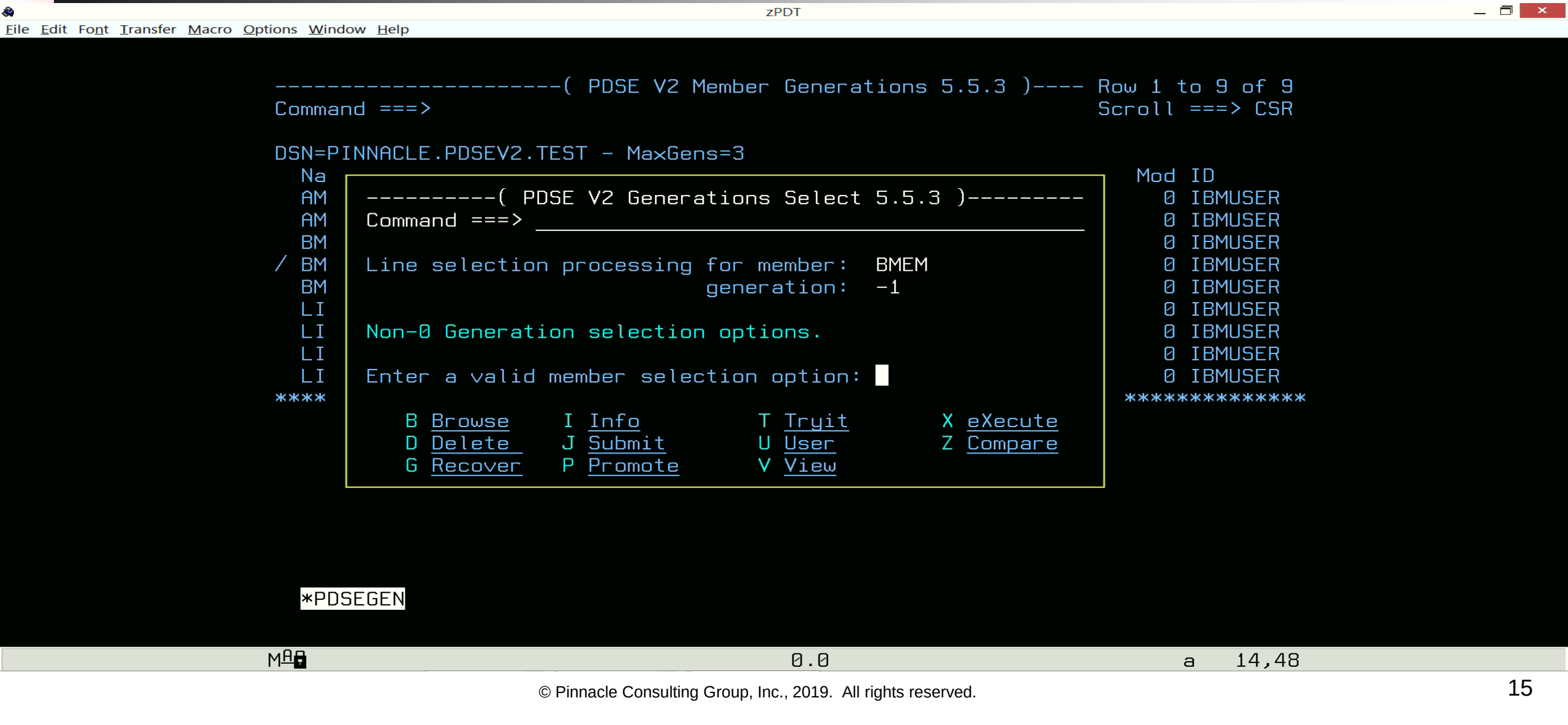
PDSEGEN Line Commands



PDSEGEN Line Commands



PDSEGEN Line Commands



PDSEGEN Line Commands for Base Members

- For base members, PDESGEN provides the following options:
  - A (Attrib) - Display Version, Mod Level, and Userid for member
  - B (Browse) - Invoke ISPF Browse for member
  - C (Copy) - Copy base member and all its generations to another PDSE
  - D (Delete) - Deletes base and all gens or just base, promoting gens
  - E (Edit) - Invoke ISPF Edit for the member (base only)
  - I (Info) - Display ISPF stats, if available, for member
  - J (Submit JCL) - Submit member to internal reader
  - K (Clone) - Copy only base member (not the gens) to new name in same PDSE
  - Q (Rename Swap) - Swaps base member names and all generations
  - R (Rename) - Renames base member and all generations

PDSEGEN Line Commands for Base Members

- For base members, PDESGEN provides the following options (cont'd):
  - T (Tryit) - Use Tryit to execute member (Rexx, CLIST, panel, skel, JCL, ASM)
  - U (User) - Invoke user command ("/" in command replaced with 'dsn(mem)')
  - V (View) - Invoke ISPF View for member
  - X (eXecute) - Executes member (CLIST/Rexx), gen executed from temp dataset
  - Z (Compare) - Compare base to a particular gen

PDSEGEN Line Commands for Member Gens

- For generations, PDESGEN provides the following options:
  - B (Browse) - Invoke ISPF Browse for member generation
  - D (Delete) - Deletes member generation
  - **NOTE NO EDIT FOR NON-ZERO GENS, CORRUPTS PDSE - OA54890 **
  - G (Recover) - Copy member generation to new base member name
  - I (Info) - Display ISPF stats, if available, for member generation
  - J (Submit JCL) - Submit member generation to internal reader
  - P (Promote) - Promote member gen to base member, base becomes -1 gen (same as SAVE NEWGEN, but safer and will not corrupt PDSE)
  - T Tryit - Use Tryit to execute member gen (Rexx, CLIST, panel, skel, JCL, ASM)
  - U User - Invoke user command ("/" in command replaced with 'dsn(mem)')

PDSEGEN Line Commands for Member Gens

- For generations, PDESGEN provides the following options(cont'd):
  - V (View) - Invoke ISPF View for member generation
  - X (eXecute) - Executes member gen (CLIST/Rexx) from temp dataset
  - Z (Compare) - Compare gen to base or other gens

PDSEGEN Miscellaneous Primary/Batch Commands

- BACKUP primary command will backup PDSE and retain gen info
- PDSEGBAK Rexx exec will BACKUP/RESTORE PDSE in batch
- PDSEGENC Rexx exec will copy PDSE in batch, retaining gens in copy
- PRUNE command deletes obsolete member generations
- VALIDATE command invokes IEBPDSE to verify PDSE
  - Can also clean PDSE with PERFORMPENDINGDELETE operand of IEBPDSE

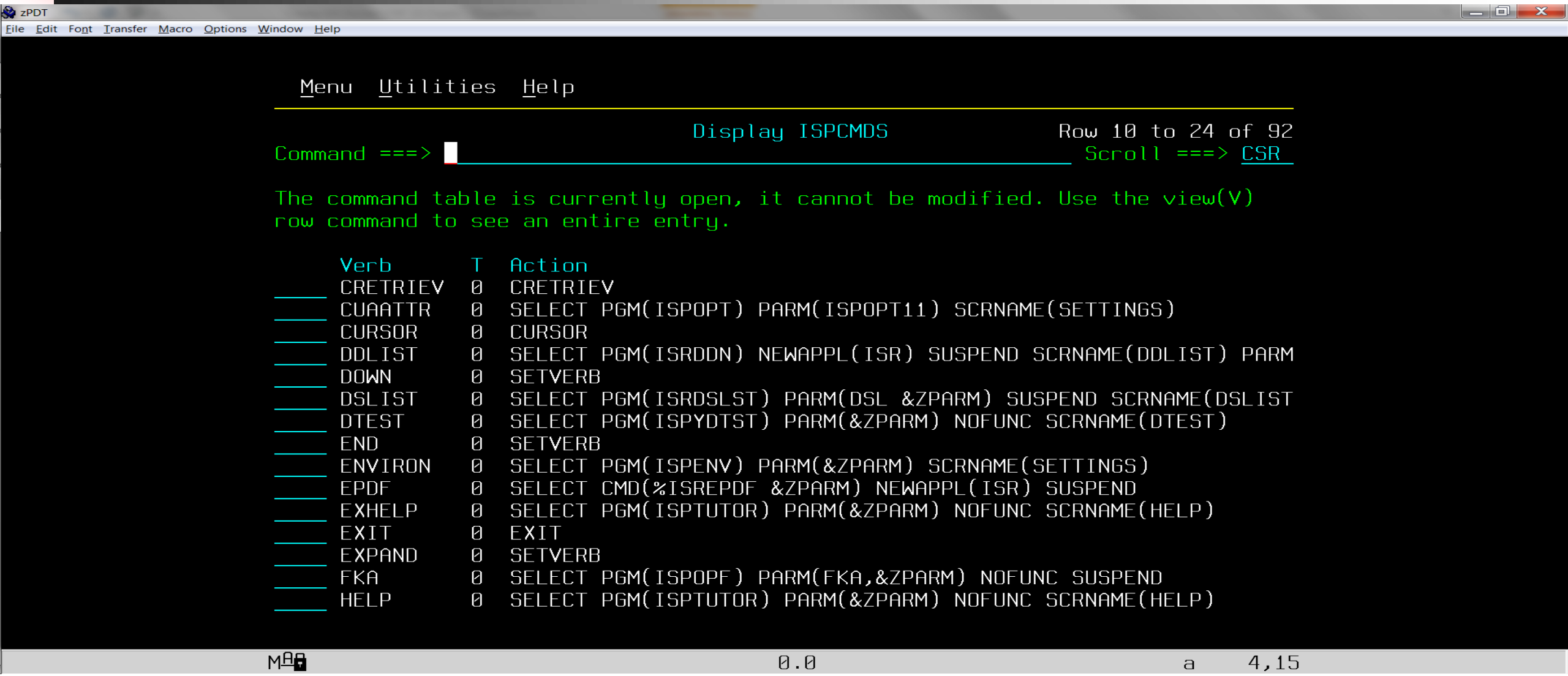
CBT Usermod Collection for ISPF (CUCI)

- In August 2017, I began to work on user-driven ISPF enhancements
- CBT Usermod Collection for ISPF (CUCI) was born
- Download File967 at http://www.cbttape.org
- All enhancements via usermod, no manual hacks allowed

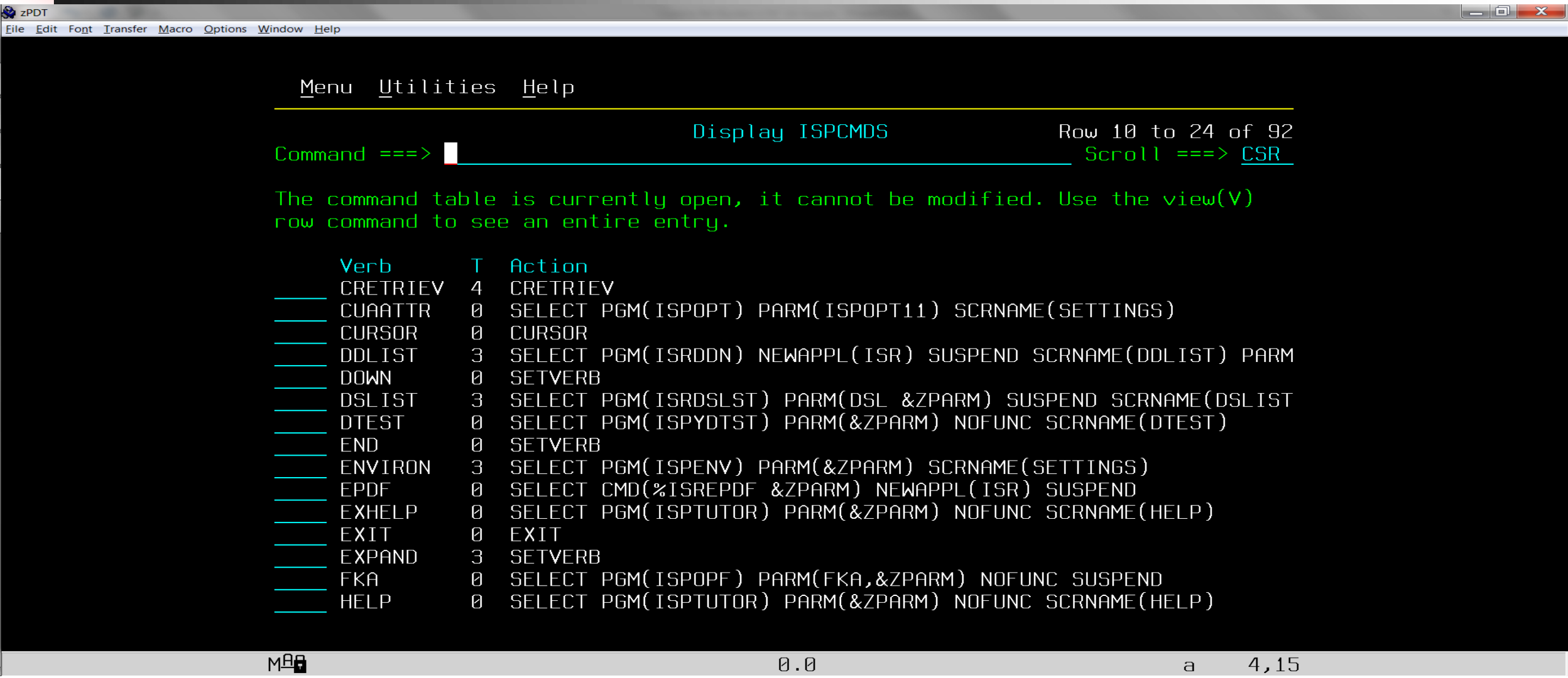
CUCI Release V1.0

- First release V1R0 delivered on 11/16/2017, small proof of concept
- Satisfied two RFE's
  - RFE 112749 - Update ISPCMDS table entries for Minimum TRUNC values (e.g. UDLIST should be 3)
  - RFE 112745 - Add DSL as a synonym for the DSLIST command
- Created ISPCMDS table member with abbreviated commands

Default ISPCMDS



Updated ISPCMDS



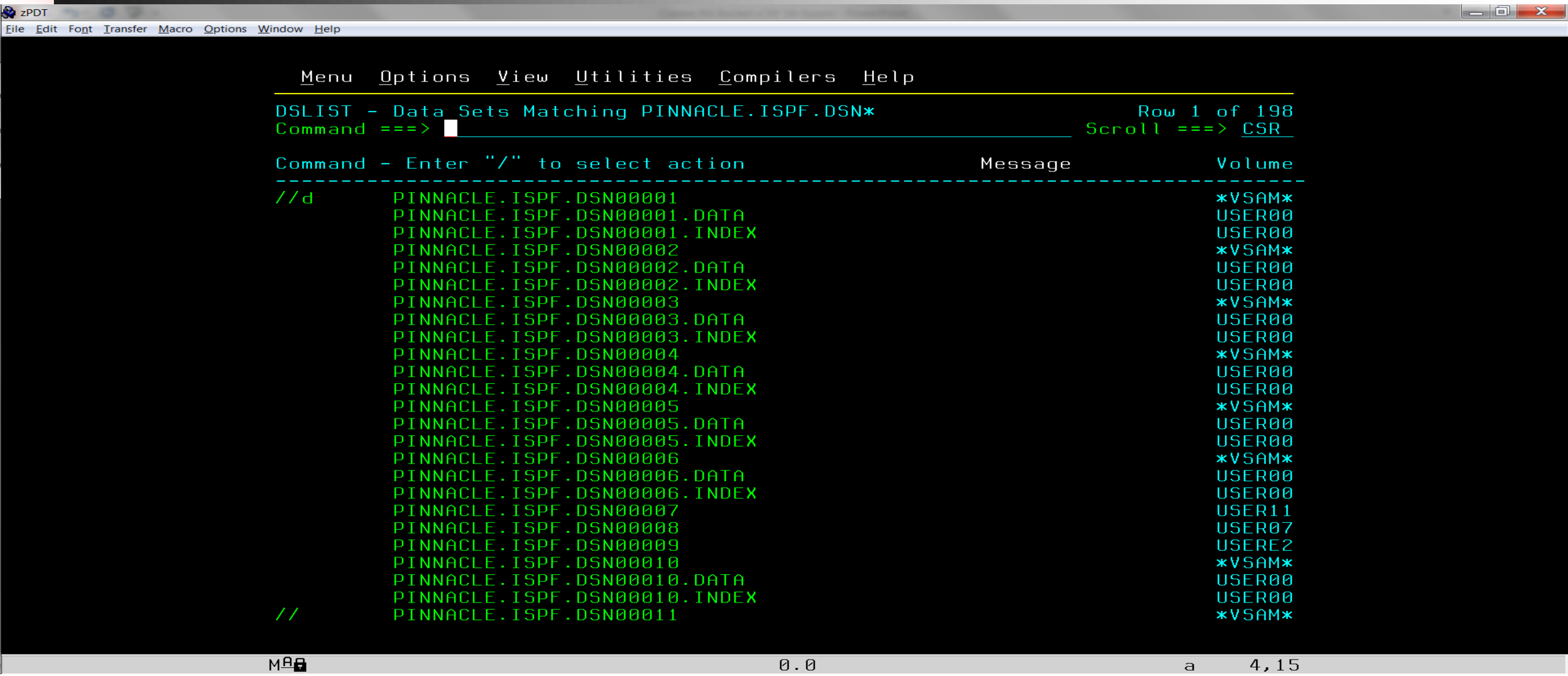
ISPCMDS Usermod (No Hacks Allowed!)

```
//IBMUSERU JOB 'IBMUSER',CLASS=A,NOTIFY=&SYSUID,
//             MSGCLASS=X,REGION=0M,TIME=NOLIMIT,MSGLEVEL=(1,1)
//*************************************************************
//APPLYUM  EXEC PGM=GIMSMP
//SYSPRINT DD SYSOUT=*
//SMPCSI   DD DISP=SHR,DSN=yourhlq.GLOBAL.CSI  <=== YOUR ISPF CSI
//CBTFILE  DD DISP=SHR,DSN=yourhlq.CBT.FILE967 <=== CBT FILE
//SMPCNTL  DD *
 SET BDY(GLOBAL).
   RECEIVE S(UMISPCM) SYSMODS.
 SET BDY(target).   /* <=== YOUR TARGET ZONE */
   APPLY S(UMISPCM) CHECK.
   APPLY S(UMISPCM).
/*
//SMPPTFIN DD DATA,DLM=@@
++USERMOD(UMISPCM) REWORK(20173190).   /*<=== RESET REWORK IF NEEDED */
++VER (Z038) FMID(HIF7R02). /*<=== YOUR ISPF FMID (ADD PRE IF NEEDED)*/
++TBLENU(ISPCMDS) DISTLIB(AISPTENU) TXLIB(CBTFILE).
@@
```

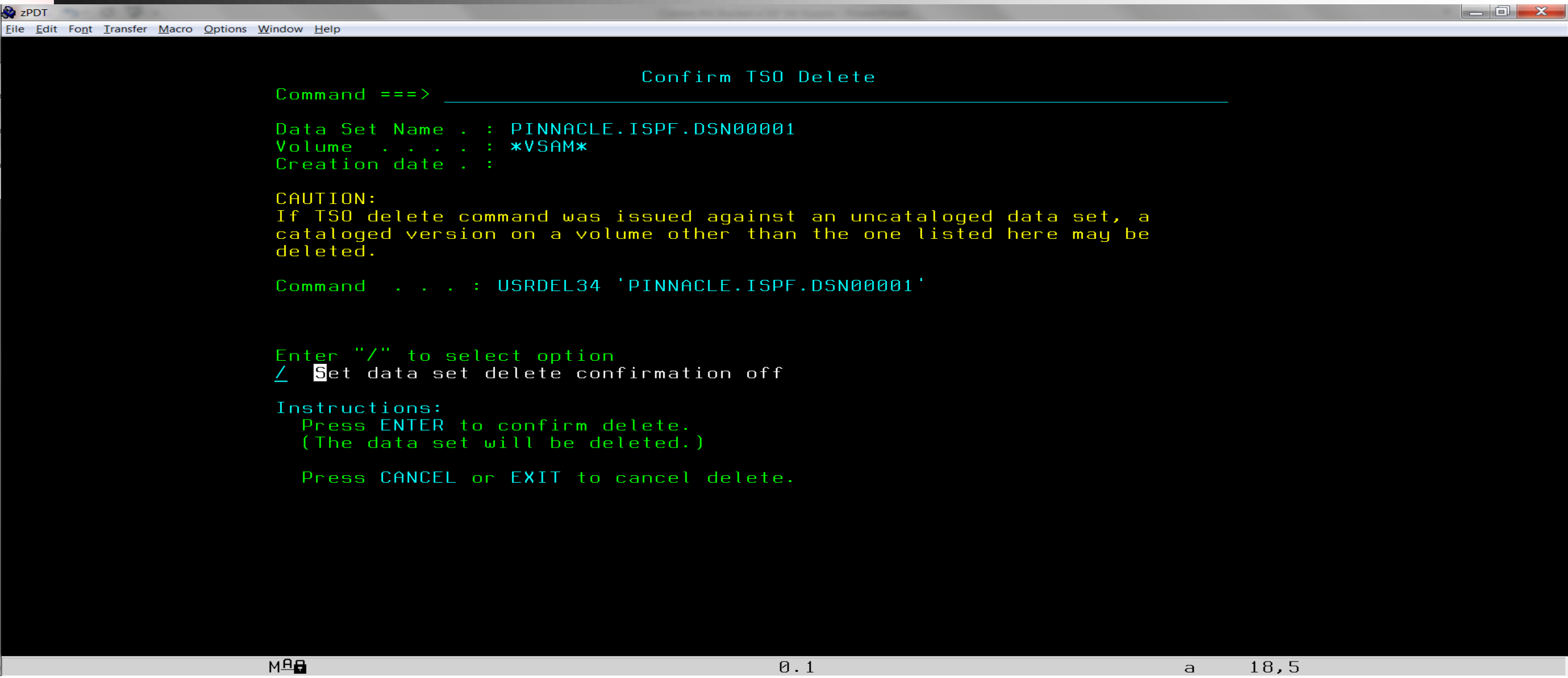
CUCI Release V1R1

- During 2018, attempted two other ISPF enhancements (and 1 z/OS)
  - RFE 128561 - ISPF OPT 3.4 block delete commands should delete VSAM cluster, and bypass DATA and INDEX components without stopping
  - RFE 120120 - ISPF 3.17 Should Reset UID on Exit Back to User UID from SU
  - RFE 102407 - ISPF UDLIST - for reset to users default UID on exit (same as above RFE 120120)
  - Modified ISHELL to reset user's default UID on exit (z/OS usermod)
- Experienced multiple coding difficulties
- Finally delivered on 12/24/2018 (My 2018 Christmas gift to you!)

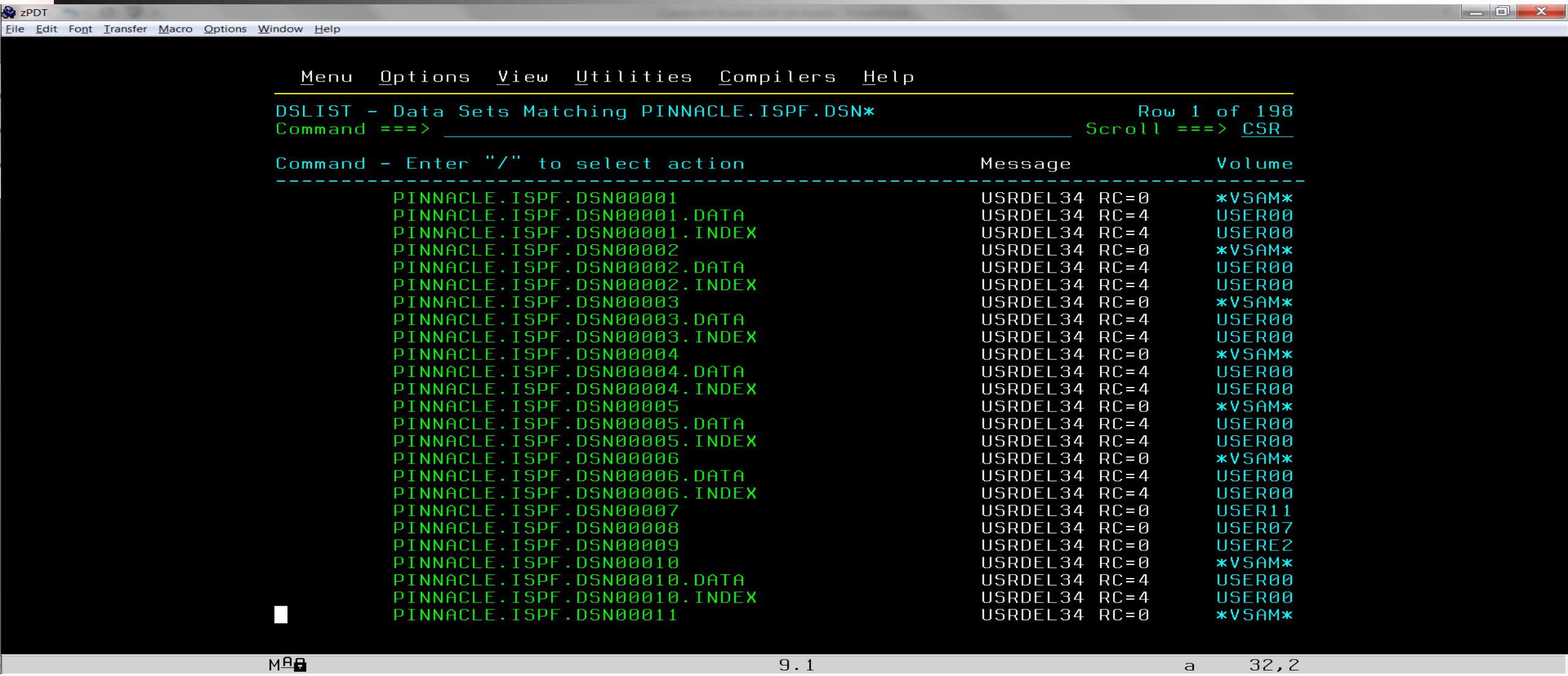
Block DELETEs for VSAM in 3.4 Without Stopping



Block DELETEs for VSAM in 3.4 Without Stopping



Block DELETEs for VSAM in 3.4 Without Stopping



Block DELETEs for VSAM in 3.4 Without Stopping

- Rexx exec USRDEL34 invokes IDCAMS and suppresses messages
- Any errors from IDCAMS set RC=4 to bypass ISPF error handling
- Note RC=4 for DATA and INDEX components
- Panel ISRUDSL0 ignores errors, bypasses dataset, and continues
- RACF and I/O errors will also be bypassed
- Because any errors are bypassed, you may have some cleanup
- After block delete, issue REFresh command to see updated list

Block DELETEs for VSAM in 3.4 Without Stopping

- Panel Rexx in ISRUDSL0)PROC injecting USRDEL34 command

```
do i = 1 to #lines
   endpos   = i * 80
   startpos = endpos - 79
   parse var zdata begzdata =(startpos) scrnline =(endpos) rstzdata
   parse upper var scrnline 1 attr 2 command 6 13 rstline
   if command = '//D ' then
      zdata = begzdata      ||,
              attr          ||,
              '//USRDEL34 ' ||,
              rstline       ||,
              rstzdata
end
```

Block DELETEs for VSAM in 3.4 Without Stopping

- USR34DEL exec:

```
/*********************************************************************/
/* Parse the arguments for the dataset name to be deleted.           */
/*********************************************************************/
parse arg dsn
/*********************************************************************/
/* VGET the delete confirmation variable ZDSLCONF.  It the value is  */
/* not equal to 'OFF', then display the delete confirmation panel,   */
/* otherwise set the delrc variable to 0 and proceed with DELETE.    */
/*********************************************************************/
address ispexec "VGET (ZDSLCONF)"
```

Block DELETEs for VSAM in 3.4 Without Stopping

```
if zdslconf <> 'OFF' then
   do
      address ispexec "DISPLAY PANEL(ISRUADC4)"
/*********************************************************************/
/* Save the return code from the delete confirmation panel display.  */
/* If the user hit 'ENTER' on the panel, rc will be 0, any other key */
/* will be a non-zero rc.                                            */
/*********************************************************************/
      delrc = rc
   end
else
   delrc = 0
```

Block DELETEs for VSAM in 3.4 Without Stopping

```
/*****************************************************************/
/* If the user hit 'ENTER' to confirm the delete, or if delete   */
/* confirmations were off, allocate files and invoke IDCAMS to   */
/* process the DELETE.                                           */
/*****************************************************************/
if delrc = 0 then
   do
/*****************************************************************/
/* Dummy out SYSPRINT to suppress any IDCAMS messages coming back to */
/* the terminal.  Allocate SYSIN, write the DELETE command to it,    */
/* set the lastrc to 0, call IDCAMS to complete the DELETE, and FREE */
/* SYSPRINT AND SYSIN.                                           */
/*****************************************************************/
```

Block DELETEs for VSAM in 3.4 Without Stopping

```
address tso "ALLOC FI(SYSPRINT) DUMMY REUSE"
     address tso "ALLOC FI(SYSIN) UNIT(SYSALLDA) SPACE(1) TRACK",
                         "RECFM(F B) LRECL(80)"
     sysin.1 = " DELETE "dsn
     "EXECIO * DISKW SYSIN (OPEN STEM SYSIN. FINIS)"
     lastrc = 0
     address tso "CALL *(IDCAMS)"
```

Block DELETEs for VSAM in 3.4 Without Stopping

```
/*****************************************************************/
/* Set all non-zero return codes to 4, to echo back a small error to */
/* the user.  The return code 8 that IDCAMS receives when trying to  */
/* delete any DATA or INDEX components causes a misleading "Command  */
/* failed" message to appear on the re-display of ISRUDSL0.  Setting */
/* the return code to 4 prevents this, while still showing the user  */
/* "USRDEL34 RC=4" in the message area.                              */
/*****************************************************************/
      if rc <> 0 then
         lastrc = 4
      address tso "FREE FI(SYSPRINT)"
      address tso "FREE FI(SYSIN)"
   end
```

Block DELETEs for VSAM in 3.4 Without Stopping

```
/*****************************************************************/
/* Set all non-zero return codes to 4, to echo back a small error to */
/* the user.  The return code 8 that IDCAMS receives when trying to  */
/* delete any DATA or INDEX components causes a misleading "Command  */
/* failed" message to appear on the re-display of ISRUDSL0.  Setting */
/* the return code to 4 prevents this, while still showing the user  */
/* "USRDEL34 RC=4" in the message area.                              */
/*****************************************************************/
      if rc <> 0 then
         lastrc = 4
      address tso "FREE FI(SYSPRINT)"
      address tso "FREE FI(SYSIN)"
   end
```

Block DELETEs for VSAM in 3.4 Without Stopping

```
/*********************************************************************/
/* If the user hit 'CANCEL' or 'END' on the delete confirmation      */
/* panel to cancel the delete, set lastrc = 4.                       */
/*********************************************************************/
else
   lastrc = 4
/*********************************************************************/
/* Send back the return code to prevent the misleading "Command      */
/* failed" error message, and keep the re-display of ISRUDSL0 clean. */
/*********************************************************************/
exit(lastrc)
```

Block DELETEs for VSAM in 3.4 Without Stopping

- Panel Rexx in ISRUDSL0)INIT to bypass errors on DELETE

```
do i = 1 to #lines
   <snip>
   parse var zdata begzdata =(startpos) scrnline =(endpos) rstzdata
   parse upper var scrnline 1 attr 2 command 11 rstline
   if command = 'USRDEL34 ' then
       do
          zdata = begzdata    ||,
                  attr        ||,
                  '          ' ||,
                  rstline     ||,
                  rstzdata
          usrdlbyp = 'BYPASS'
          leave i
       end
 end
```

Block DELETEs for VSAM in 3.4 Without Stopping

- If "bypass" flag is set, simulate enter to continue processing

```
IF (&USRDLBYP = 'BYPASS')
   .MSG = ''
   .RESP=ENTER
```

ISPF 3.17 Reset UID to Default on Exit

```
/*************************************************************/
/* The following code is added to ensure that the UID is reset to the */
/* user's default UID upon exit from 3.17.  This is necessary if the  */
/* user entered SUPERUSER mode while in 3.17.                         */
/*************************************************************/
IF (.RESP = 'END')
   *REXX(ZINIUID ZEUID)
    call syscalls('ON')
    address syscall 'seteuid' ziniuid
    address syscall 'getuid'
    zusruid = retval
    address syscall 'geteuid'
    zeuid = retval
   *ENDREXX
```

CUCI Release V1R2

- Release V1R2 delivered on 01/28/2019
  - RFE 122097 - ISPF EDIT highlighting for FORTRAN
  - RFE 117386 - ISPF EDIT highlighting for SQL
  - RFE 79228 - ISPF EDIT highlighting for SQL
  - RFE 57717 - ISPF EDIT highlighting for SQL
  - RFE 37007 - ISPF EDIT highlighting for SQL
  - RFE 107568 - ISPF EDIT highlighting for CARLa
  - RFE 83017 - ISPF EDIT highlighting for shell script
  - RFE 83016 - ISPF EDIT highlighting for Java

CUCI Release V1R2

- Release V1R2 delivered on 01/28/2019 (continued)
  - RFE 112741 - Simplify UNIX file access in ISPF by prepending current Unix directory to filename for COMPARE, COPY, CREATE, and REPLACE commands (BROWSE, EDIT, and VIEW require member names and continue to issue "Invalid member name" error message)

EDIT Highlighting for Java, SQL, etc!

- At SHARE in San Jose, my life changed
- Pedro Vera presented Customizing the ISPF HILITE Command
- Pedro showed how to manipulate EDIT data for highlighting
- Block DELETE code showed how to manipulate panel dynamic area
- Armed with Pedro's ground-breaking work and block DELETE code, language highlighting was doable

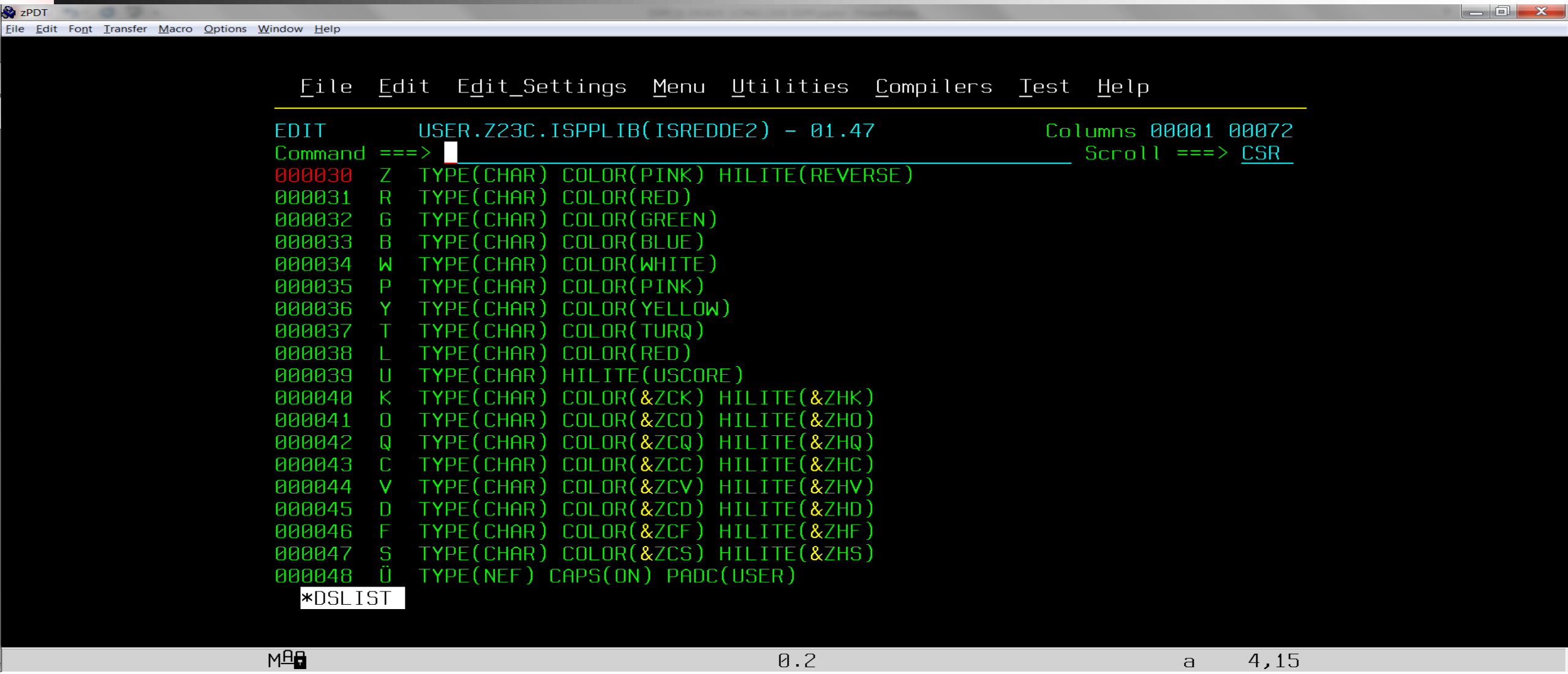
EDIT Highlighting for Java, SQL, etc!

- Highlighting starts with dynamic area of ISPF Edit panel ISREDDE2

```
26 AREA(DYNAMIC) EXTEND(ON) SCROLL(ON) USERMOD('20')
)BODY  EXPAND(//) WIDTH(&ZWIDTH)  CMD(ZCMD)
<snip>
.ZDATA,ZSHADOW/ /
2ECCEC6EECCCDE646
694131B9281466101
```

- Dynamic areas defined with data and shadow
- Shadow same length as data, assigns attribute to each byte of data
- Enables coloring at the character level

EDIT Highlighting for Java, SQL, etc!



EDIT Highlighting for Java, SQL, etc!

- Begin with ISPF command HILITE OTHER
- HILITE OTHER will handle some comments, keywords, and operators
- To highlight other languages, start by highlighting comments
  - Fortran - c, C, d, D, *, ! in col. 1 or ! in any column not part of quoted string
  - Java - any '//' characters not already in comment or quoted string
  - Python, Ruby, Shell - any '#' character not already in comment or quoted string
  - SQL - any '--' characters not already in comment or quoted string
  - Ruby - Block comments delineated by =BEGIN and =END in column 1

EDIT Highlighting for Java, SQL, etc!

- Highlighting requires blank delimiters to clearly delineate keywords
- All special characters converted to blanks (attributes, parens, etc.)

```
tempdata = zdata
tempdata = translate(tempdata,,
          '                                ',,
          '152A2B2F141B1C260102030405060708090A0B0C0D1316171D20'x)
tempdata = translate(tempdata,'                                ',,
                                '{}[]()#+-*/=<>&^|:%, ')
```

- Keywords uppercased to highlight If, Then, Do, End, etc.

```
parse upper var tempdata tempdata
```

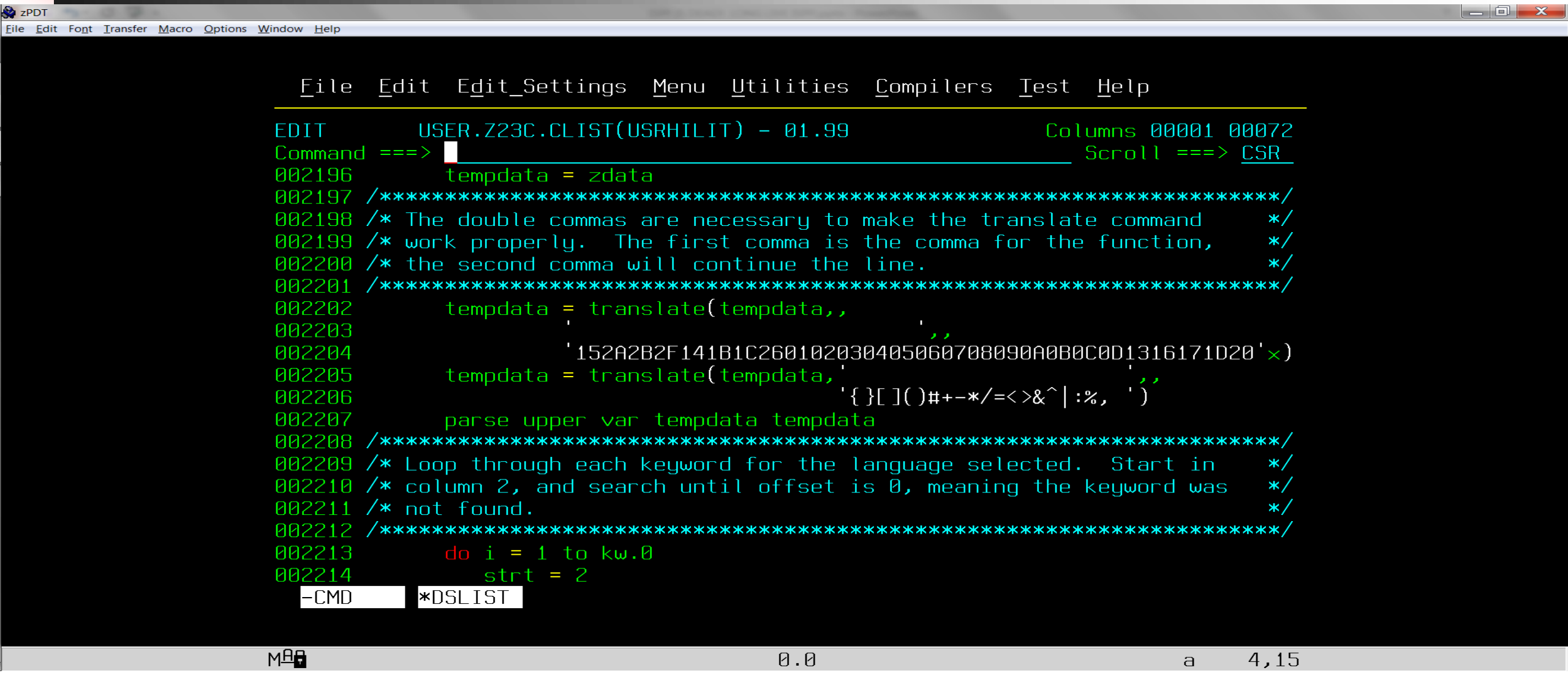
EDIT Highlighting for Java, SQL, etc!

- Modified ISREDDE2 edit panel to support new HIGHLITE command

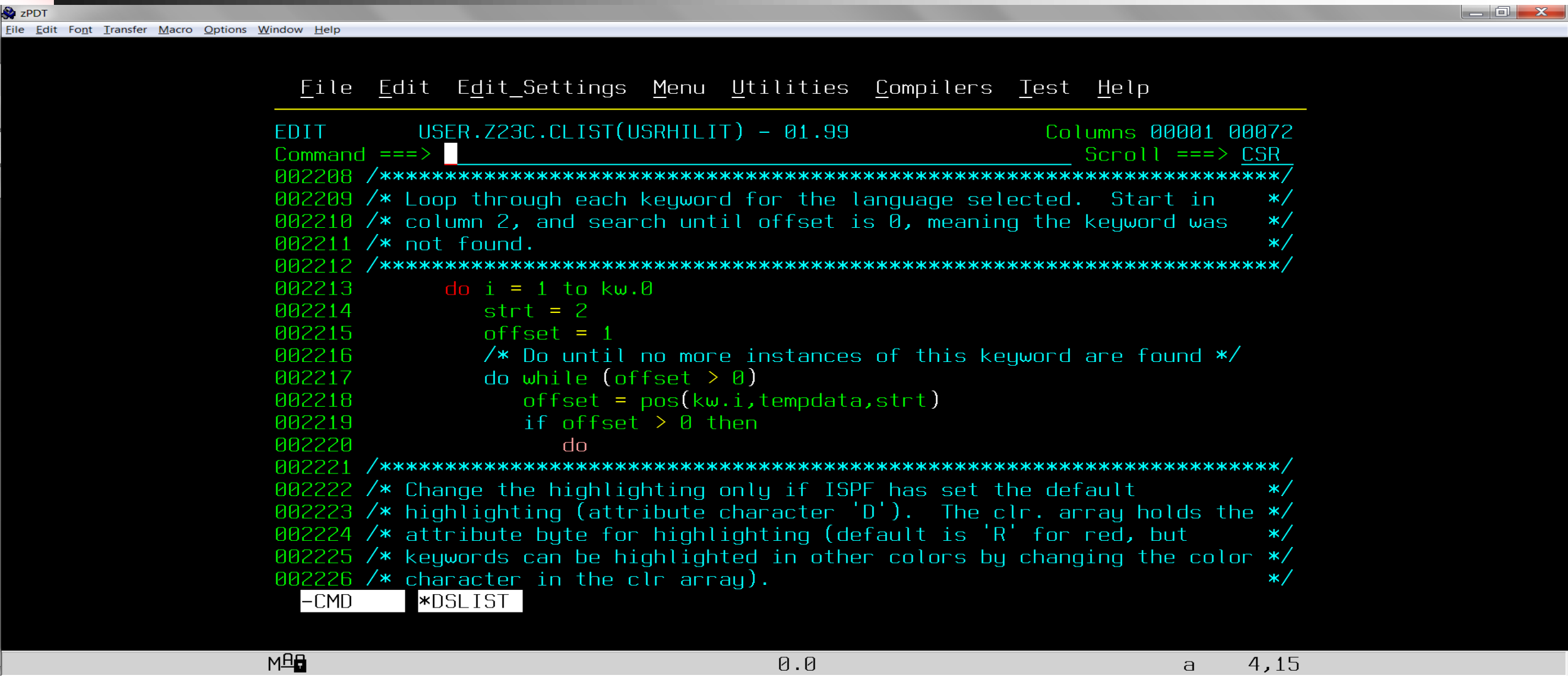
```
To use extended language highlighting, issue HILITE OTHER command
then issue the HIGHLITE command with the following syntax:

HIGHLITE language | OFF

language = CARLA | FORTRAN | JAVA | PYTHON | RUBY | SHELL | SQL
OFF turns off extended language highlighting.
HIGHLITE BRACE toggles brace highlighting on and off.
HIGHLITE BRACKET toggles bracket highlighting on and off.
HIGHLITE DISPLAY | DISP shows current state of HIGHLITE options.
HIGHLITE HELP | ? displays this HELP text.

Note:  NULLS ON ALL causes some keywords to remain unhighlighted.
       NULLS ON STD allows all keywords to be highlighted.
       Issue NULLS ON STD if not seeing all keywords highlighted.
```

Processing Comments

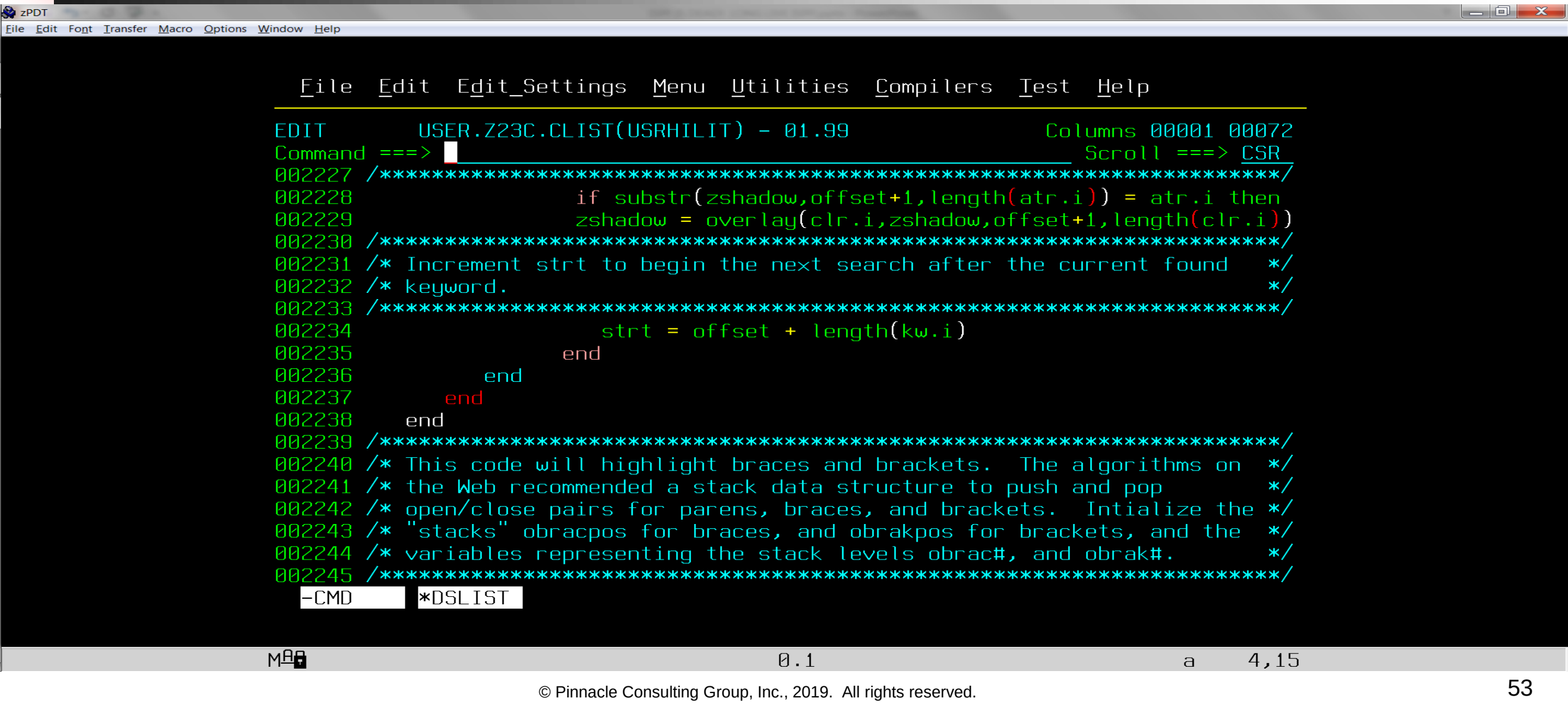
Converting Special Chars to Blanks, Uppercase



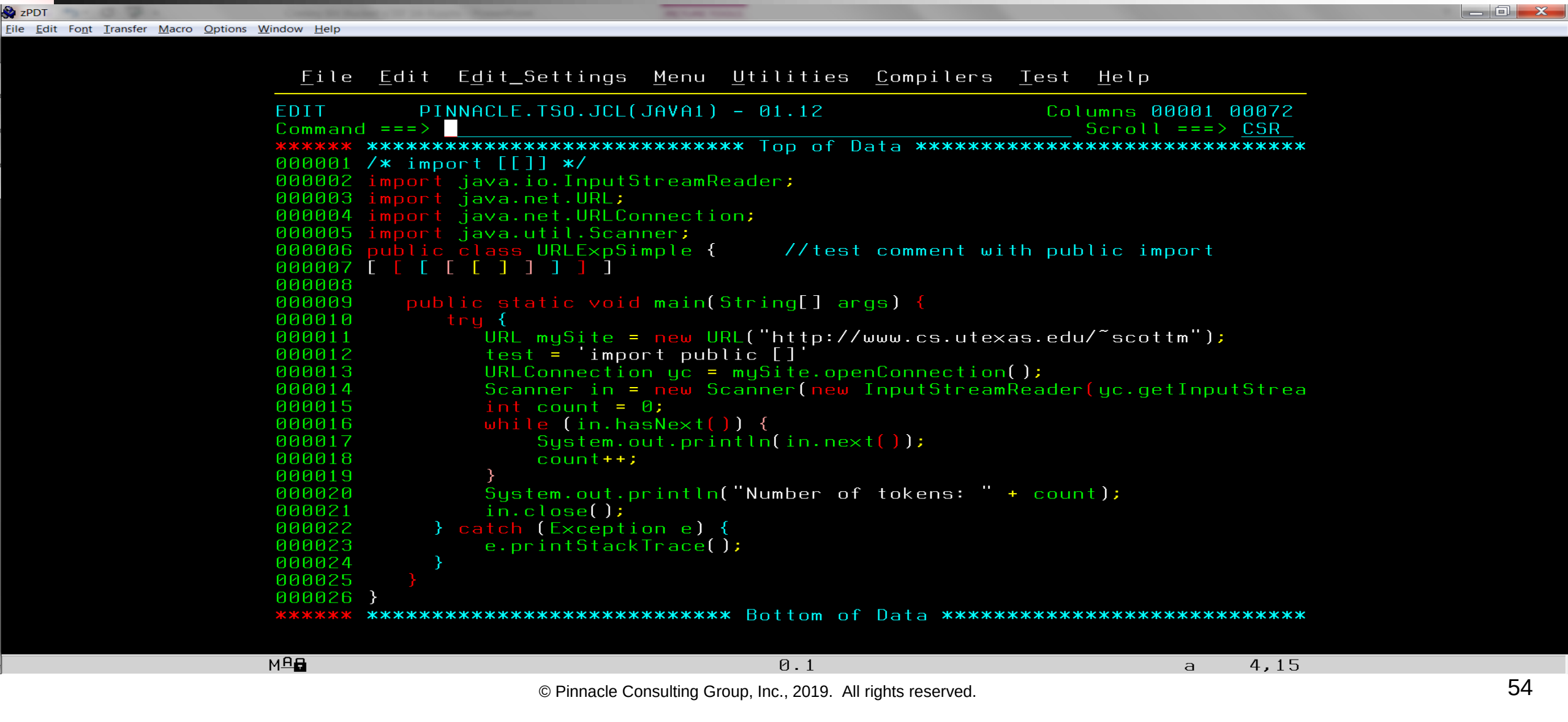
Highlighting Keywords



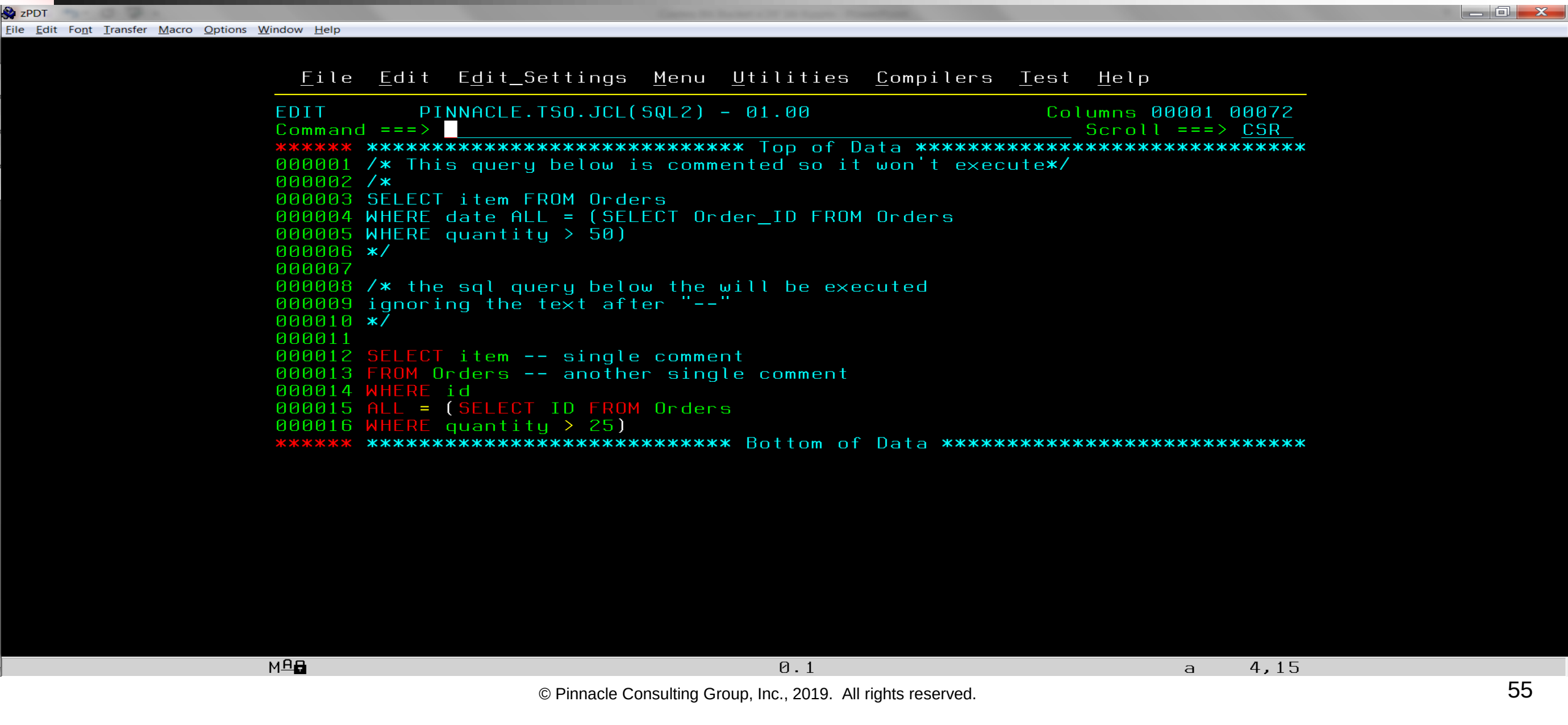
Highlighting Keywords



HIGHLITE JAVA, BRACKET, and BRACE



HIGHLITE SQL



EDIT Highlighting for Other Languages

- EDIT highlighting for other languages is now easily possible
- USRHILIT exec is easy to modify to add languages
- In addition to RFE languages, Ruby and Python also added to V1R2

Extending EDIT Highlighting - Modify USRHILIT

- USRHILIT modifications to add languages
  - Modify kw.0 variable for number of keywords

```
when (usrhilng = 'CARLA') then
  do
     kw.0    = 46
```

  - Add array of blank-delimited keywords

```
kw.1    = ' ALLOCATE '
kw.2    = ' BDAMQSAM '
```

  - Add array for default attribute string (what HILITE OTHER gives you)

```
atr.1   = 'DDDDDDDD'
atr.2   = 'DDDDDDDD'
```

  - Add array for color attribute string (color for corresponding keyword)

```
clr.1   = 'RRRRRRRR'
clr.2   = 'RRRRRRRR'
```

Extending EDIT Highlighting - Modify USRHILIT

- USRHILIT modifications to add languages (cont'd)
  - Add code to process any language-specific comments
  - This code processes '//' comments for Java

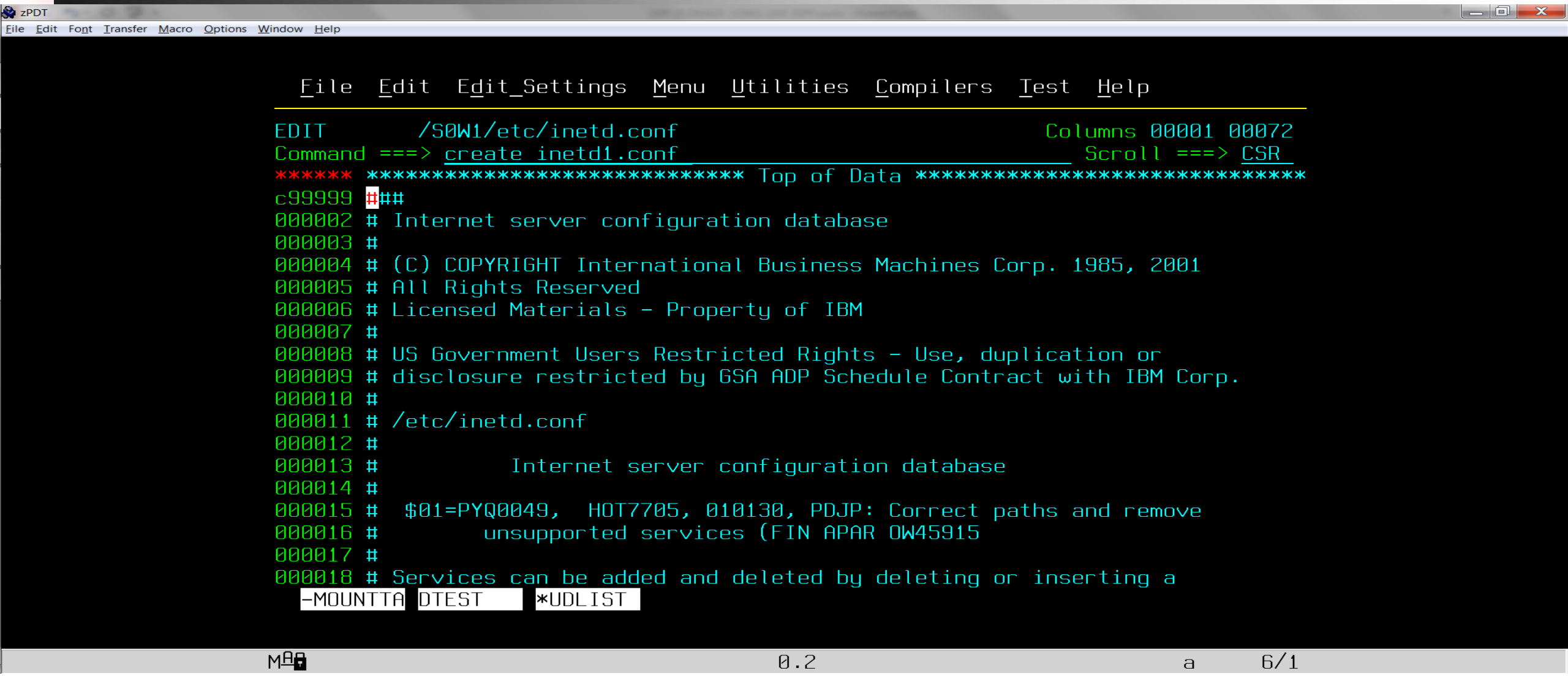
```
compos = pos('//',linedata)
if compos > 0 then
   do
      compend = compos + 2
      parse var shaddata =(compos) comshad =(compend)
      if comshad <> 'QQ' & comshad <> 'CC' then
         zshadow = overlay(,
                   copies('C',zwidth - compos),,
                   zshadow,,
                   totchar+compos,zwidth - compos)
   end
```

USS path for COMPARE, COPY, CREATE, REPLACE

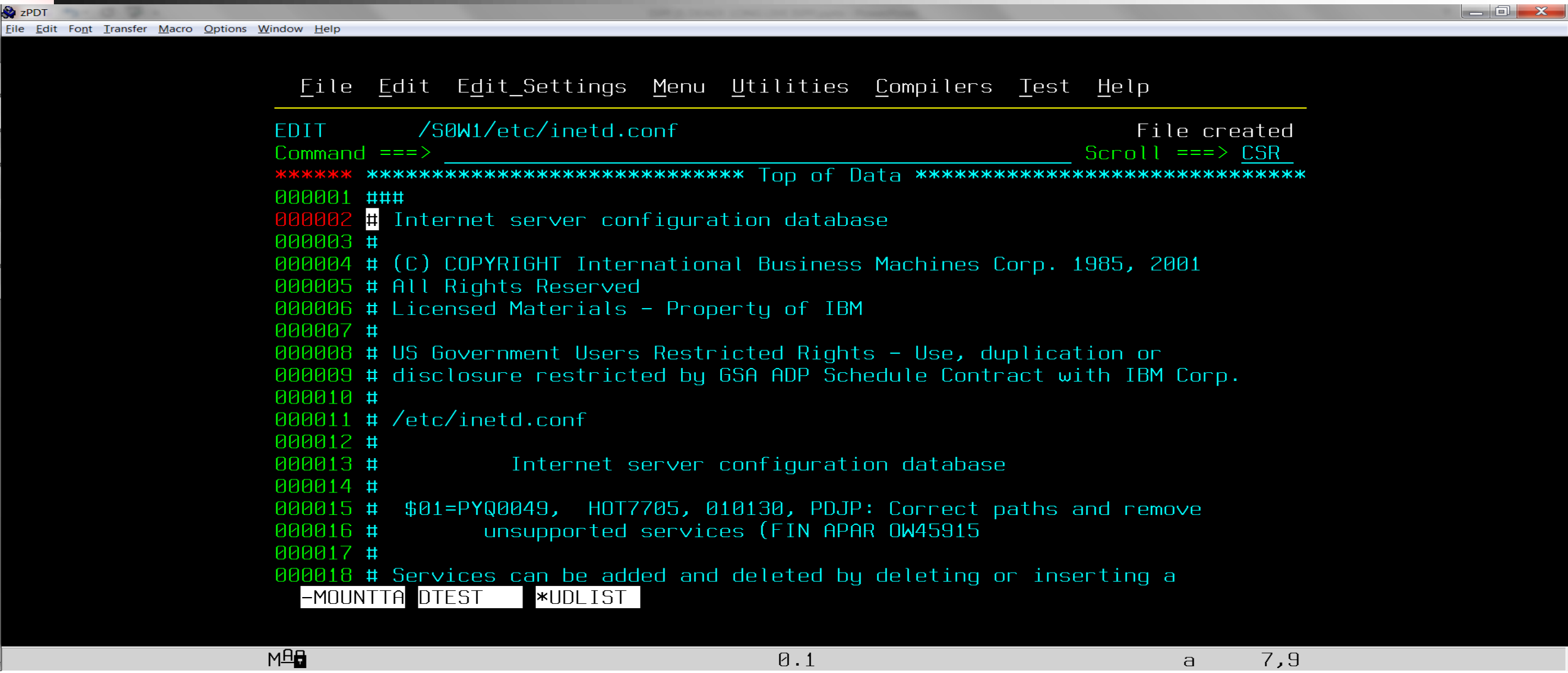
- Modified ISREDDE2 edit panel to prepend path
- Unnecessary to type full USS pathname for above commands

```
parse upper var zcmd zcmd1u zcmd2u
parse var zcmd zcmd1 zcmd2
if ((zcmd1u = 'COPY'    |,
     zcmd1u = 'COMPARE' |,
     zcmd1u = 'COMP'    |,
     zcmd1u = 'CREATE'  |,
     zcmd1u = 'CRE'     |,
     zcmd1u = 'REPLACE' |,
     zcmd1u = 'REPL')   &,
    (zpthnm <> ''       &,
     substr(zcmd2,1,1) <> '/')) then
   zcmd = zcmd1 zpthnm || '/' || zcmd2
```

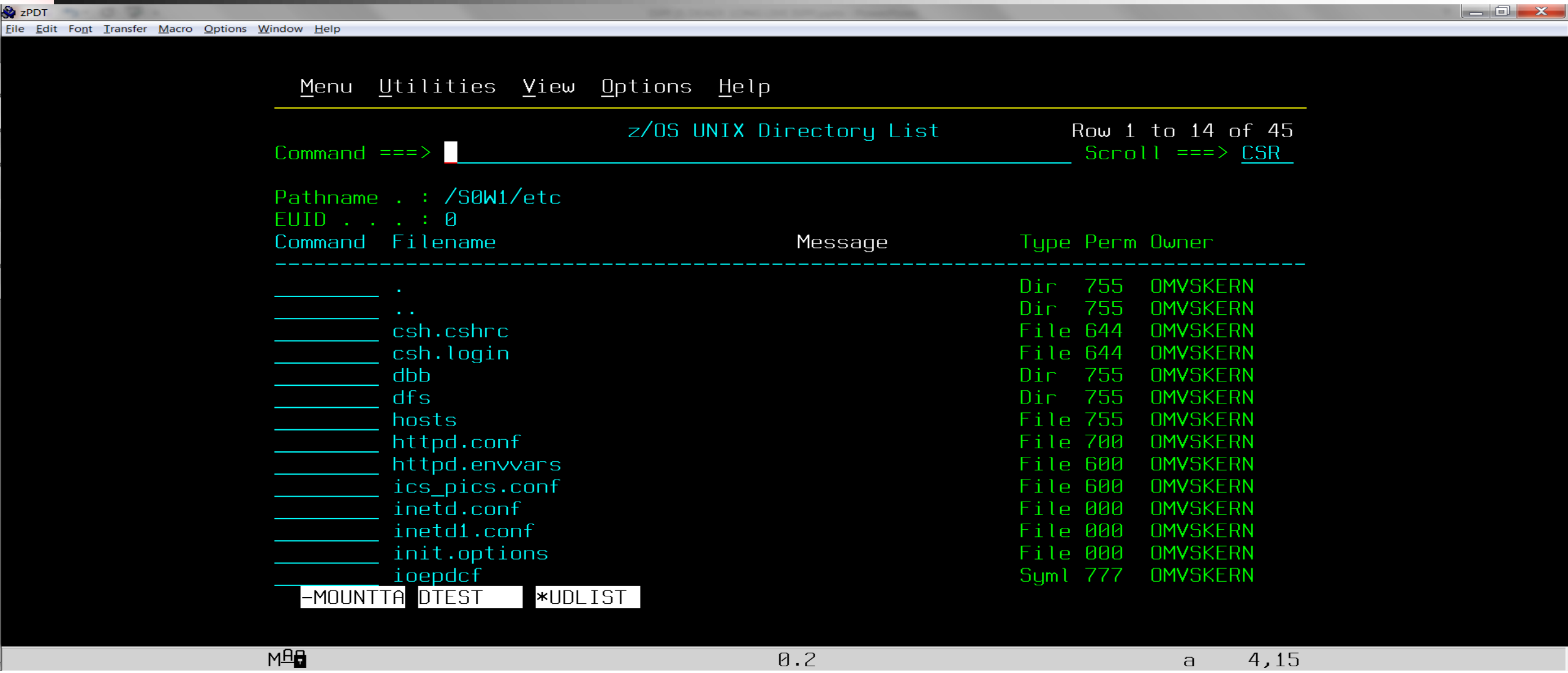
USS path for COMPARE, COPY, CREATE, REPLACE



USS path for COMPARE, COPY, CREATE, REPLACE



USS path for COMPARE, COPY, CREATE, REPLACE



CUCI Coming Attractions

- Allow ISPF Administrator to set options not in ISPCCONF
- Fix panel ISRUVPC4 to allow extended GDG's to 999
  - Panel easy to change, but ISPF doesn't add EXTENDED to IDCAMS DEFINE
  - Have to parse command in EDIT window to add EXTENDED keyword
- Additional languages to highlight
  - Lua - case sensitivity challenge
  - JavaScript
  - Swift
  - Groovy
  - Perl - so many keywords (1000+), so little time (any takers?)
  - Suggestions for other languages (Scala, R, PHP?)

CUCI Coming Attractions - Not In Plan

- Column ordering (UDLIST, FS by mount point, FS by filesystem)
  - Variables controlling column ordering are ZULARR, ZMTMARR, ZMTFARR
  - Embedded hex codes representing column, width and description
  - So far unable to translate into a usable form, it's completely obfuscated

```
ZULARR  = UDLCOLS  010D0101010402020304030604080 40A0B1305090A0A060B0C13070C0D13080D0E13...
ZMTMARR = MTBFCOLS 0108000001230101022302020304030305040404060805050703060604 0A0707080A
ZMTFARR = MTBMCOLS 0108000002230101012302020304030305040404060805050703060604 0A0707080A
```

- UDLIST (3.17) file coloring (ISHELL analog)
  - UDLIST table display, can't color rows, model attributes unchangeable
- SWAPBAR options
  - They're in ISPSPROF, not ISRPROF, so nothing we can do

Set ISPF Defaults for Options not in ISPCCONF

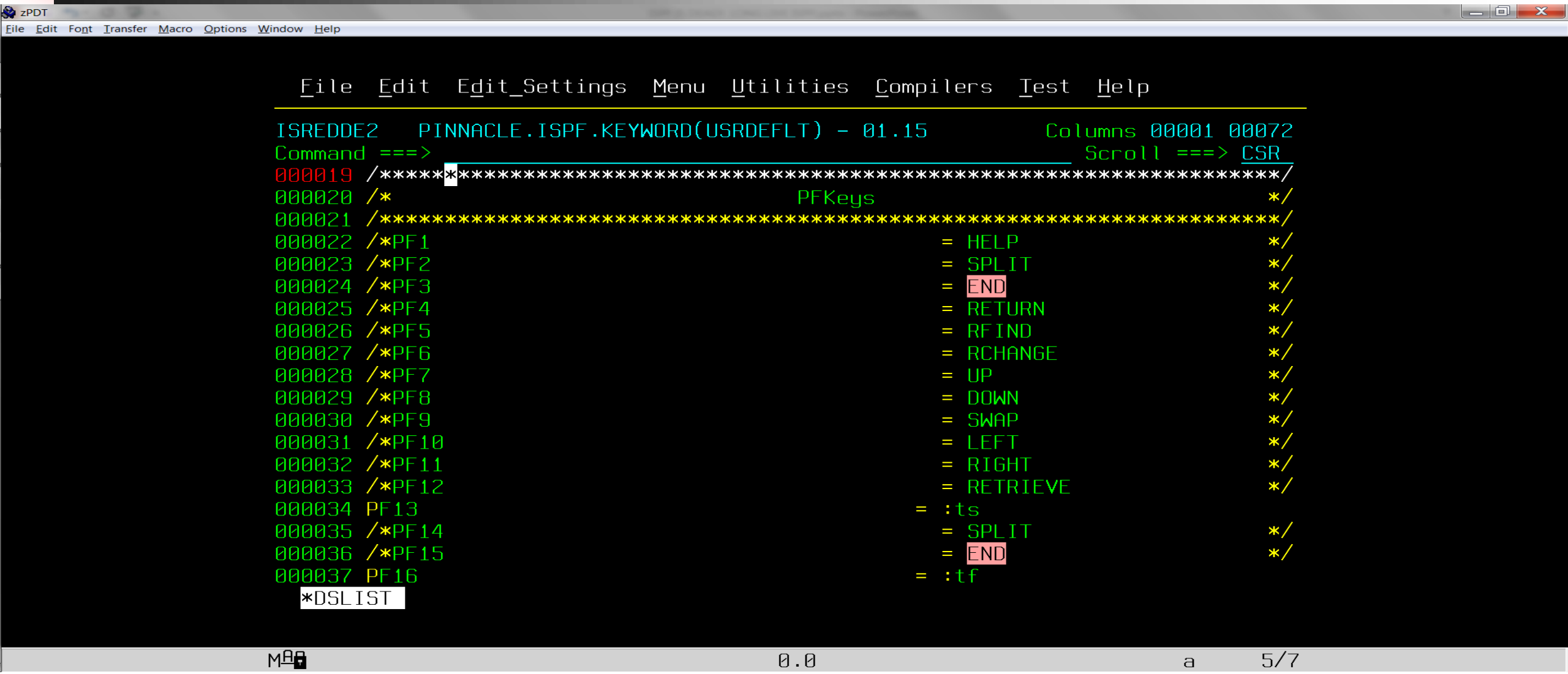
- Create keyword file member with options
- Code in ISR@PRIM to open option member and read it
- If new user, or admin specifies FORCE, update profile
- Completed items
  - PFKeys
  - MEMLIST SRCHFOR options
  - DSLIST SRCHFOR options
  - UDLIST SRCHFOR options
  - Calendar options for start day, colors, and time format
  - UDLIST options

Set ISPF Defaults for Options not in ISPCCONF

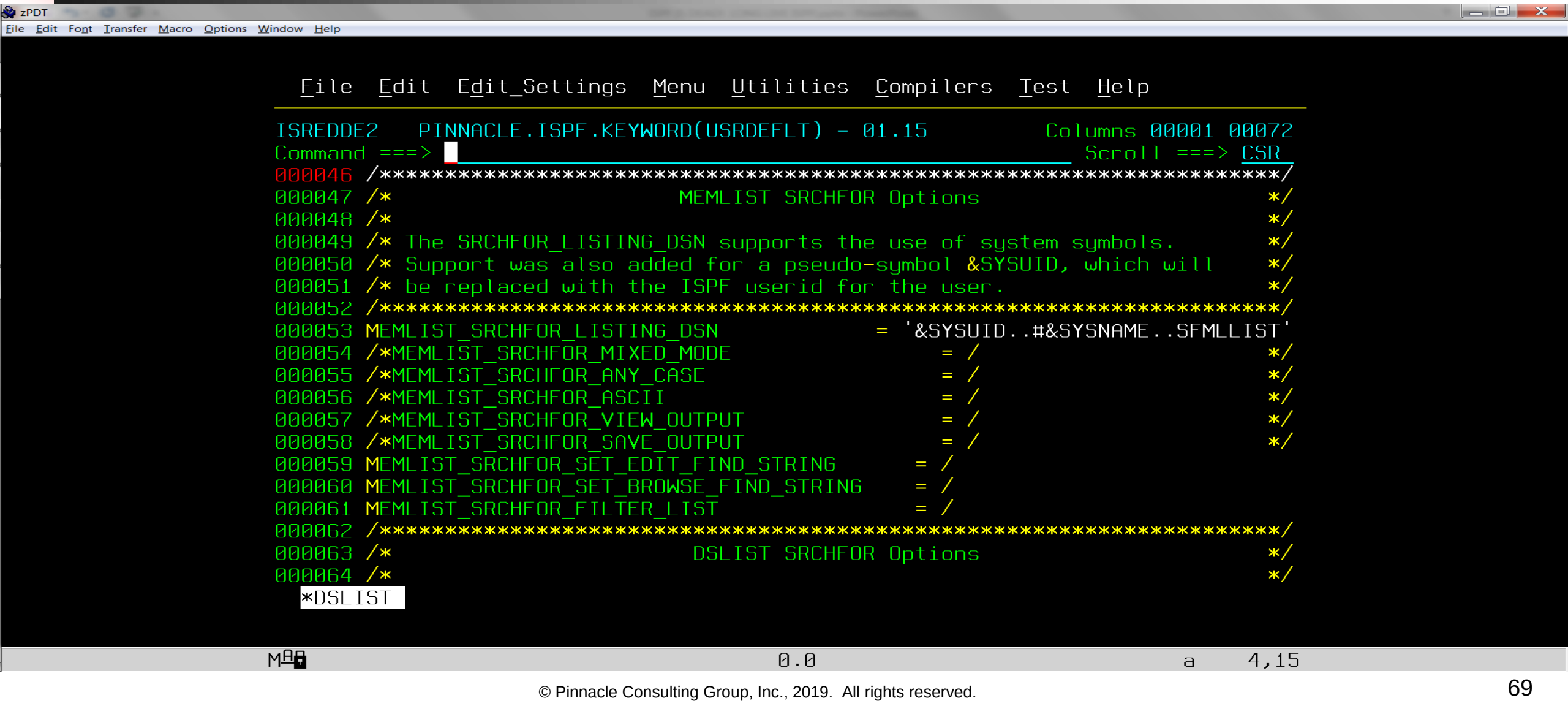
- In plan configuration items yet to be completed
  - Edit/View/Browse entry panel options
  - Mount table list options
  - z/OS UNIX Directory List Default Line Commands (Symlink "I", c'mon man!)

Set ISPF Defaults for Options not in ISPCCONF

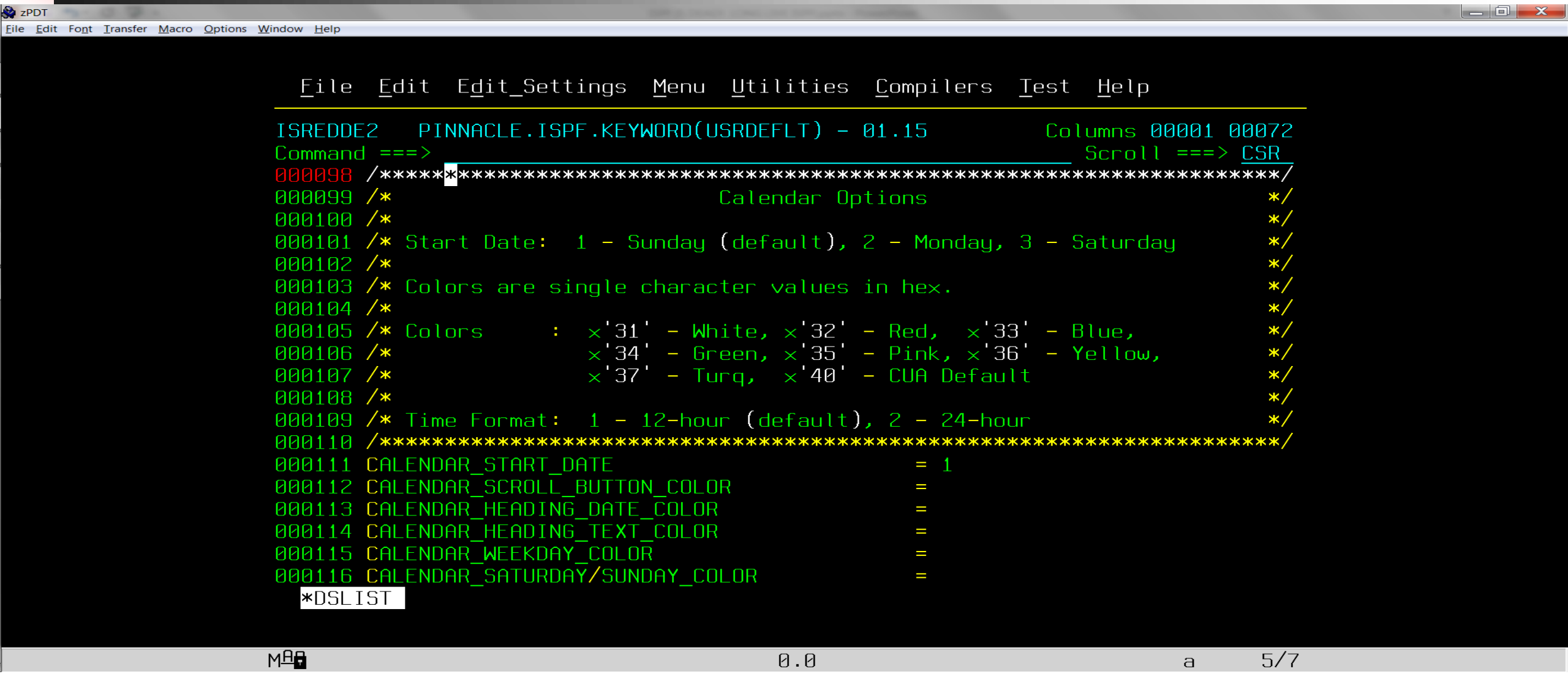
Set ISPF Defaults for Options not in ISPCCONF



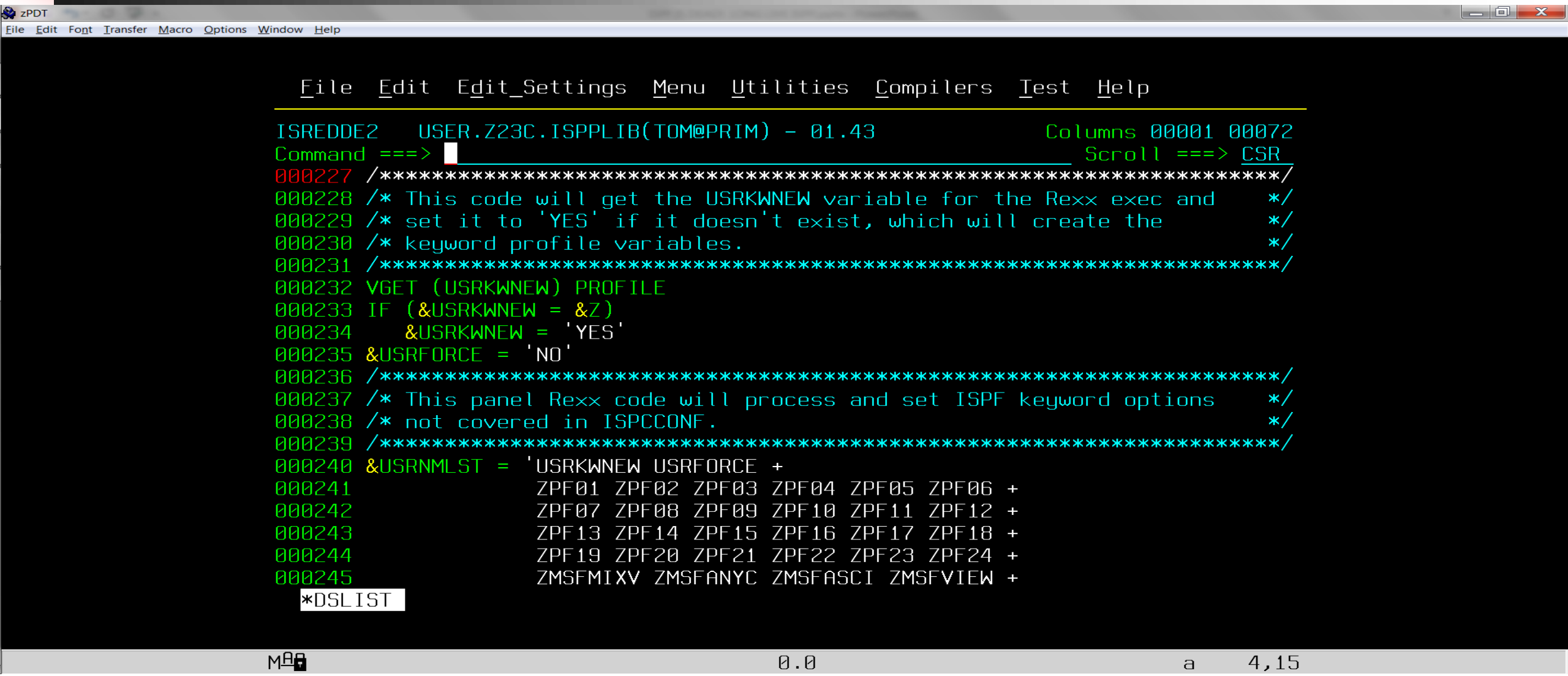
Set ISPF Defaults for Options not in ISPCCONF



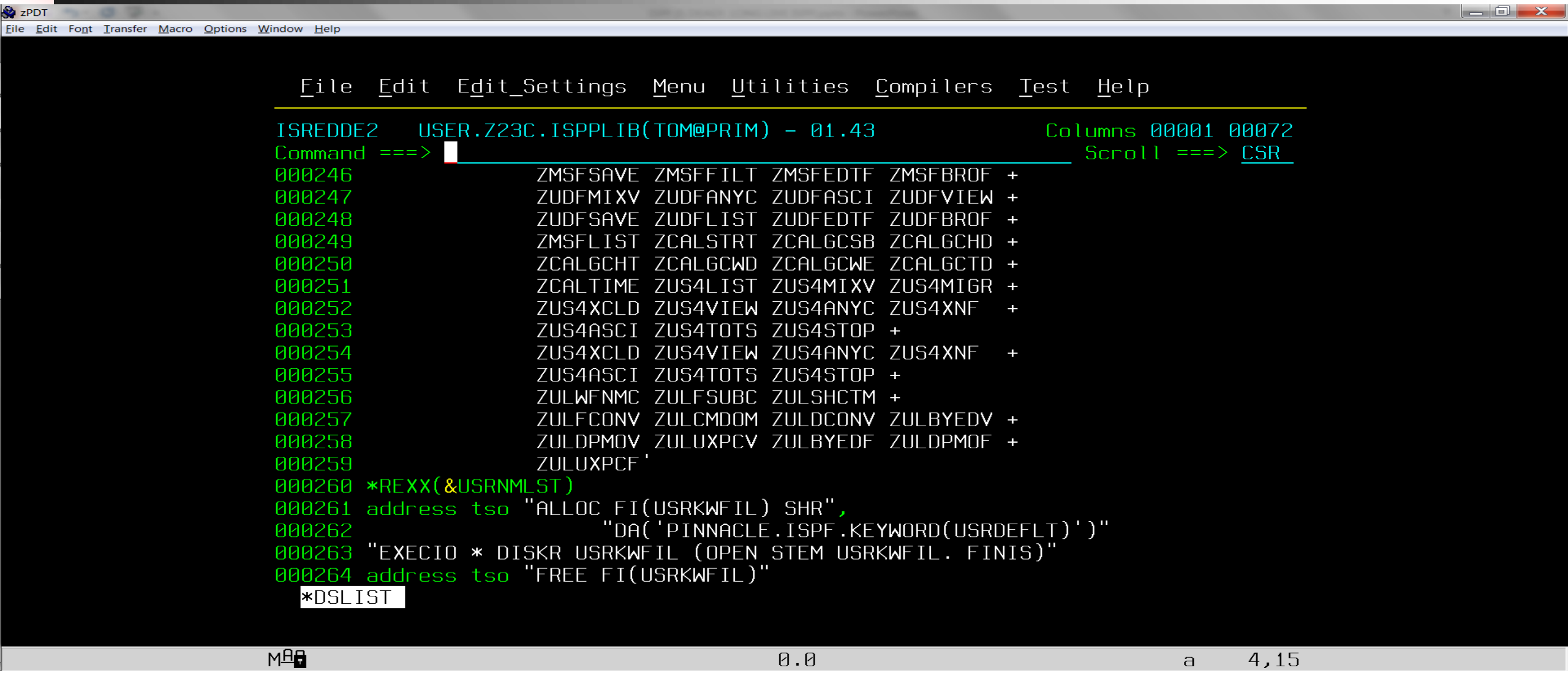
Set ISPF Defaults for Options not in ISPCCONF



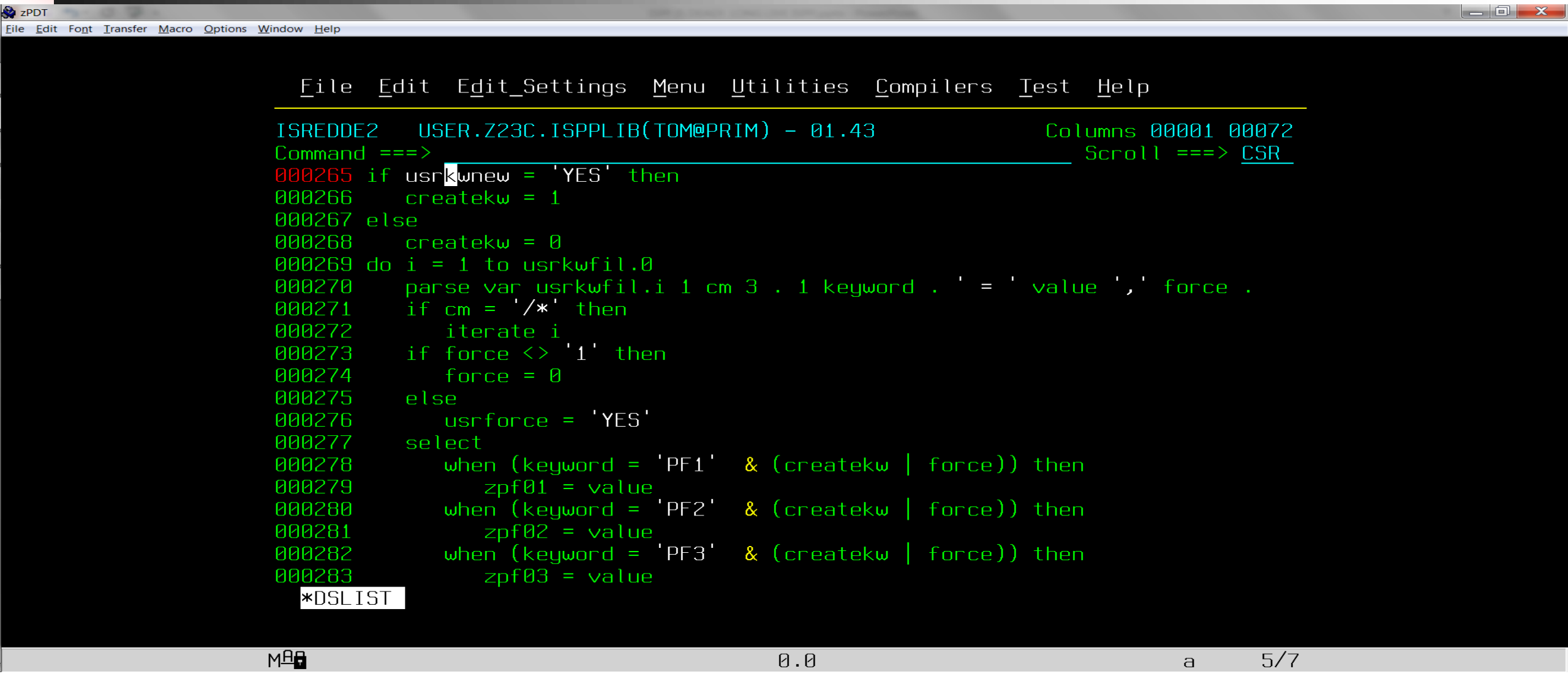
Set ISPF Defaults for Options not in ISPCCONF



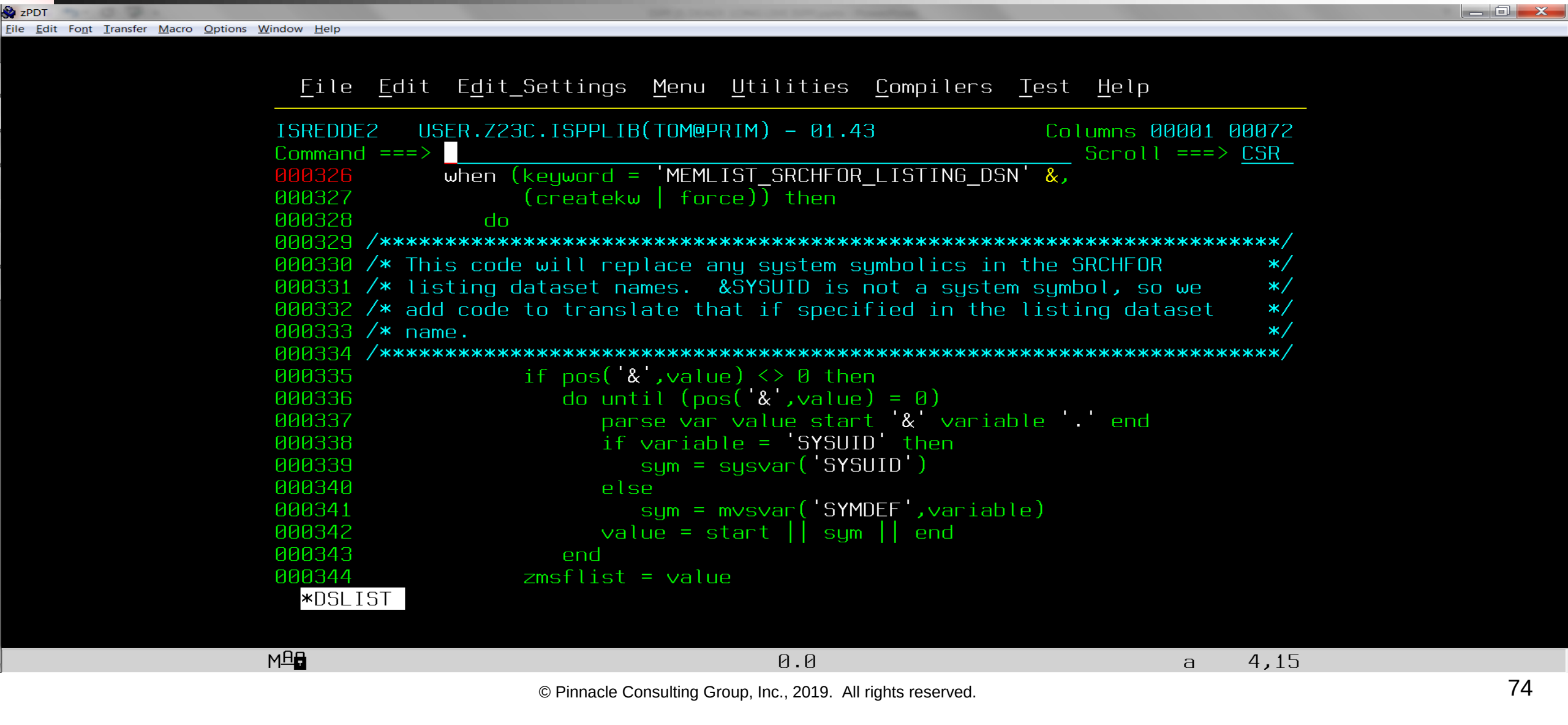
Set ISPF Defaults for Options not in ISPCCONF



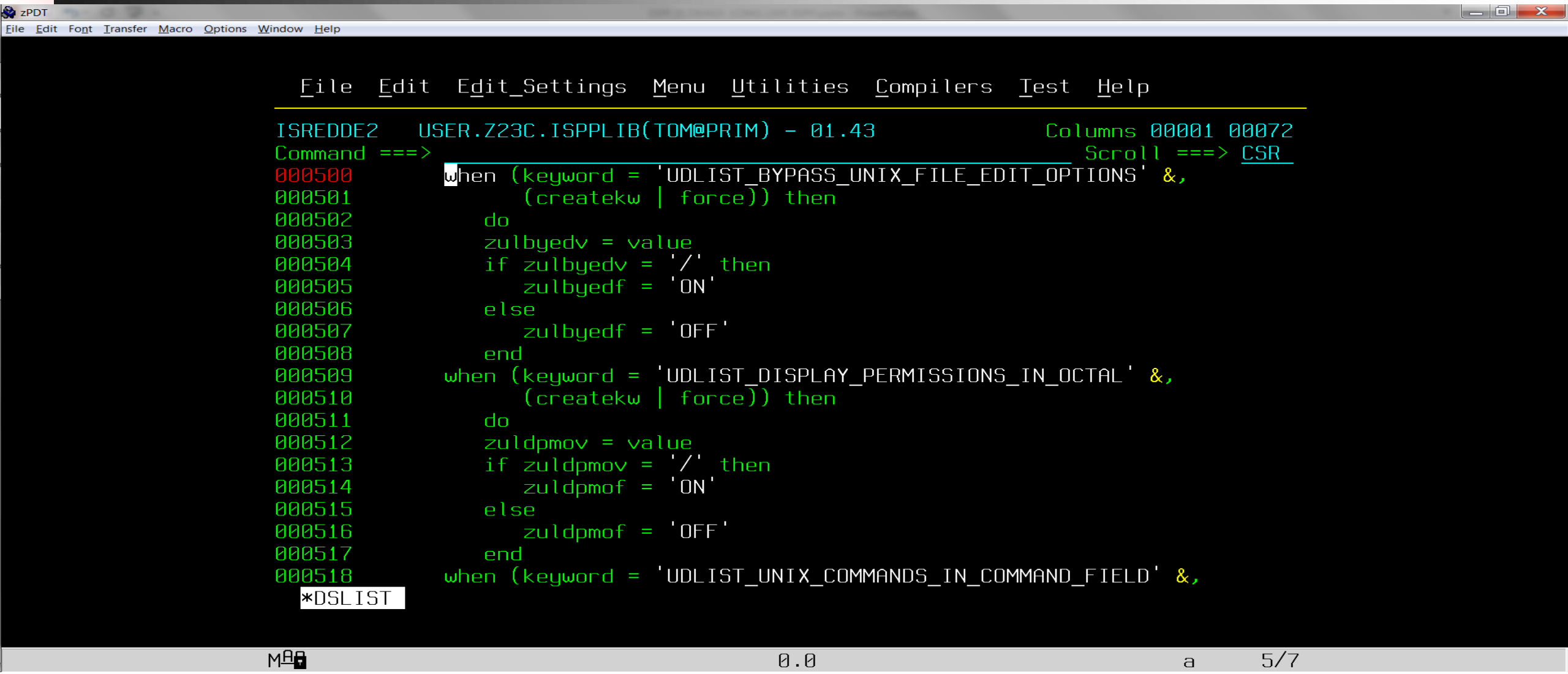
Set ISPF Defaults for Options not in ISPCCONF



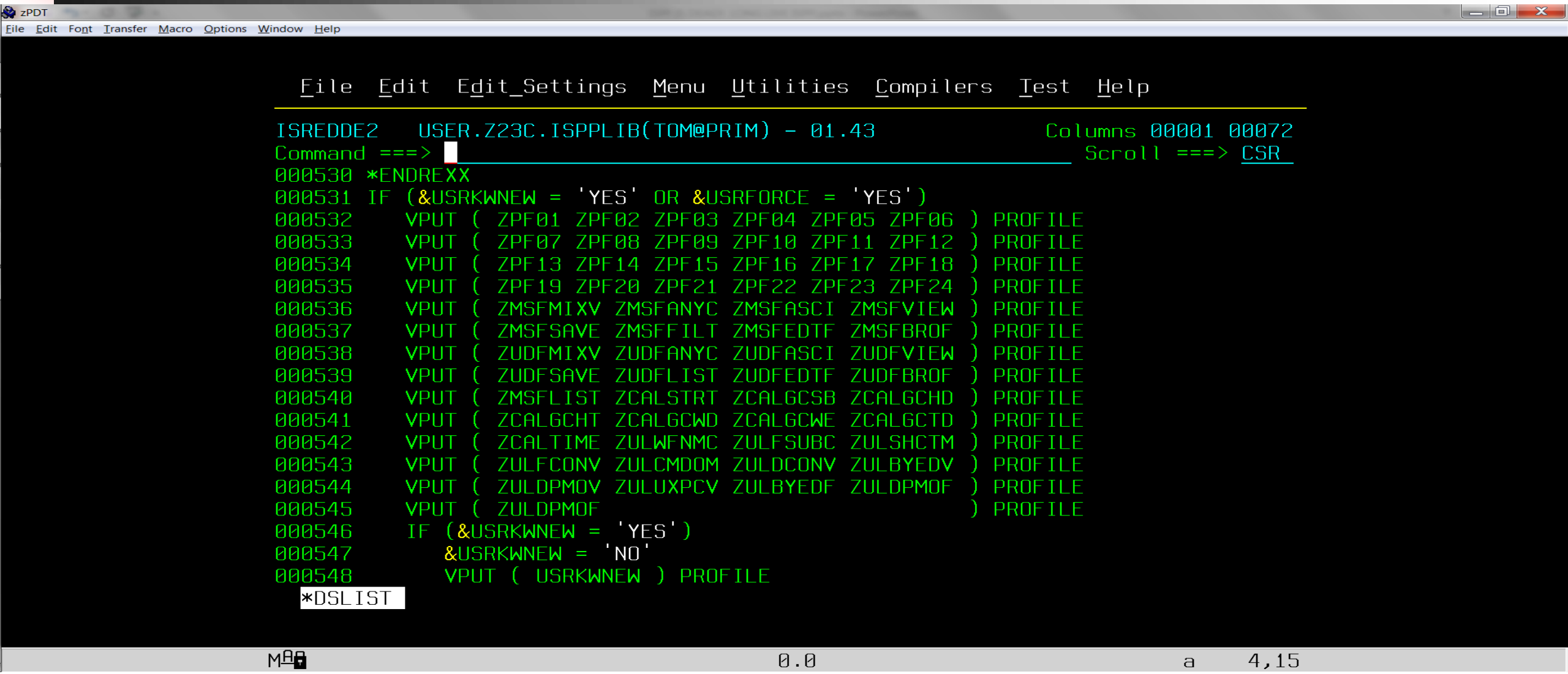
Set ISPF Defaults for Options not in ISPCCONF



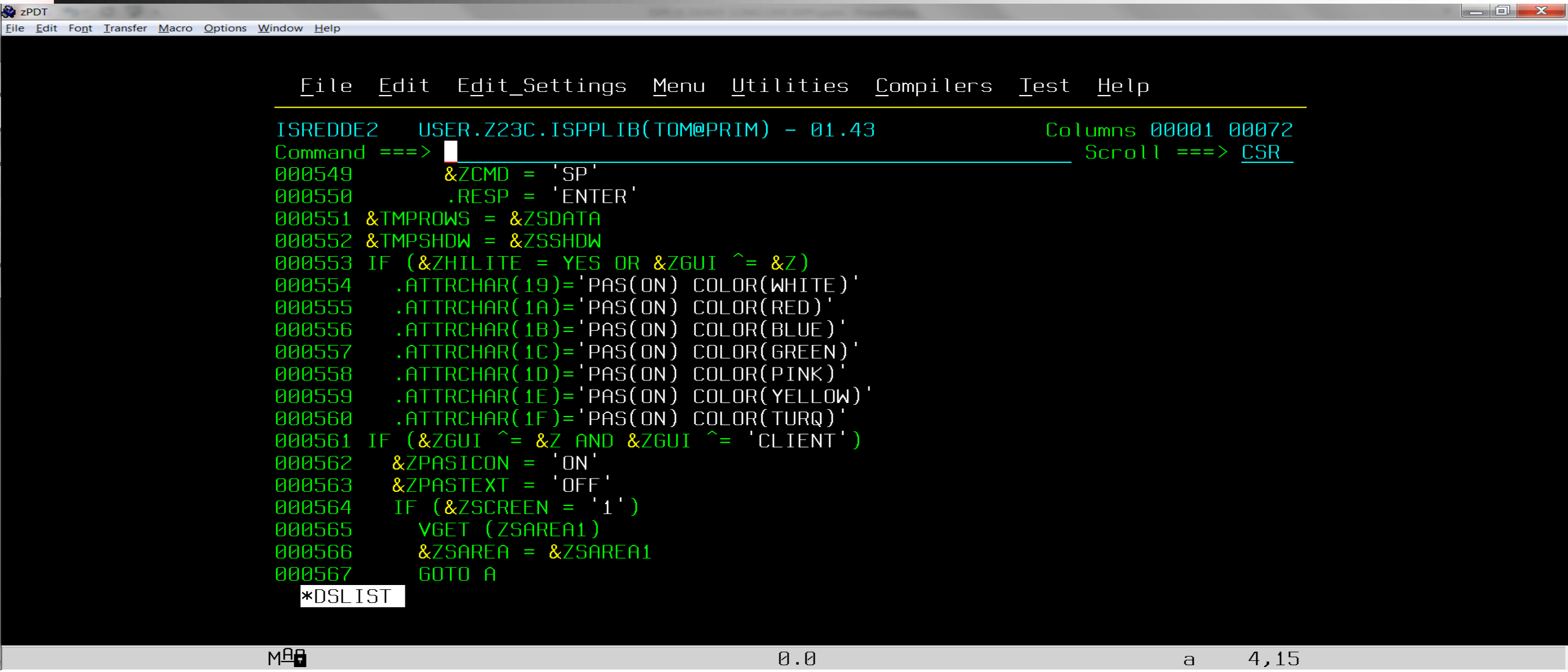
Set ISPF Defaults for Options not in ISPCCONF



Set ISPF Defaults for Options not in ISPCCONF



Set ISPF Defaults for Options not in ISPCCONF



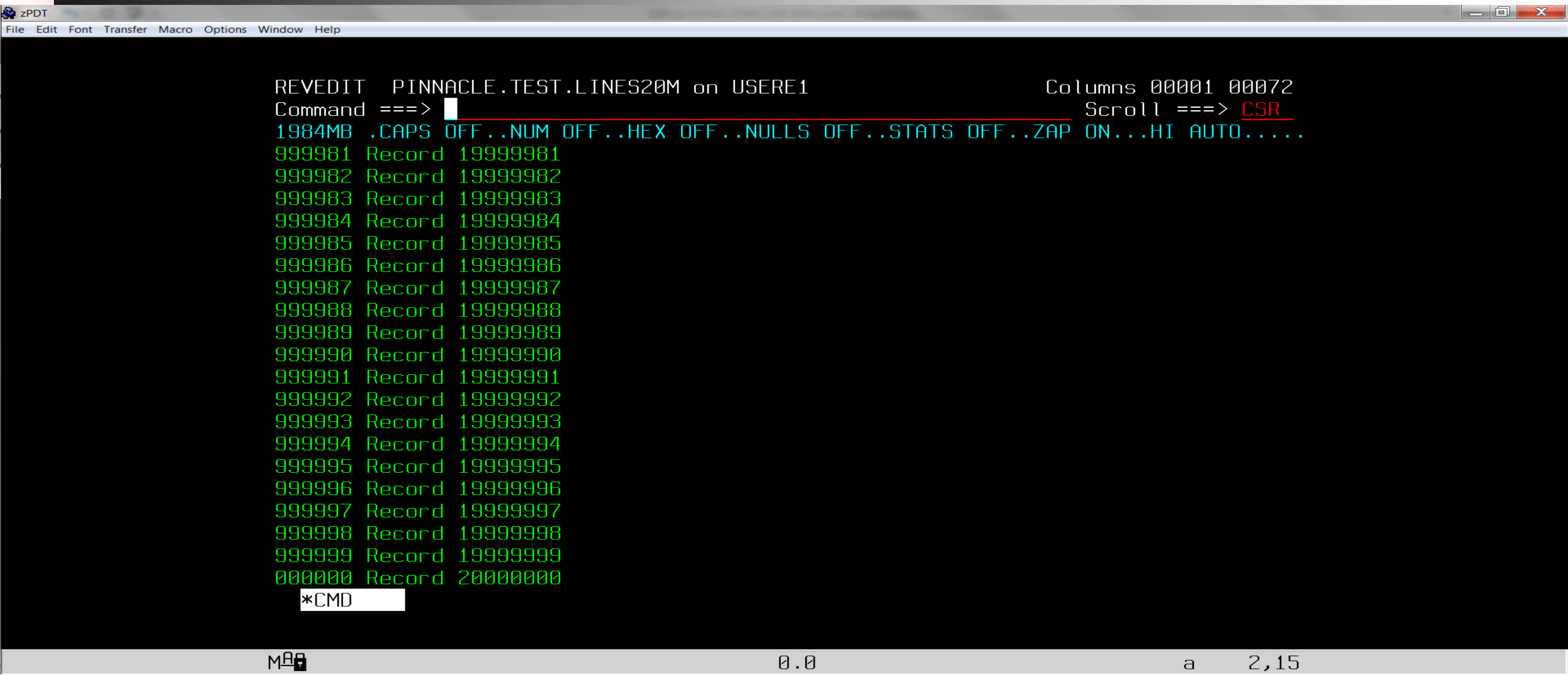
REVEDIT

- REVEDIT is part of Greg Price's REVIEW package on CBT Mods Tape
- Grab Greg's FILE134 here, and FILE135 here
- REVEDIT partially satisfies RFE 112742 for editing larger files
- Uses SCOPE-SINGLE extendable dataspace
  - Dataspace limited to 2GB or smaller by installation settings
  - Cut/Paste records held in subpool 38 in region, so can't cut 2GB of data
  - Cut/Paste does not communicate with ISPF Cut/Paste

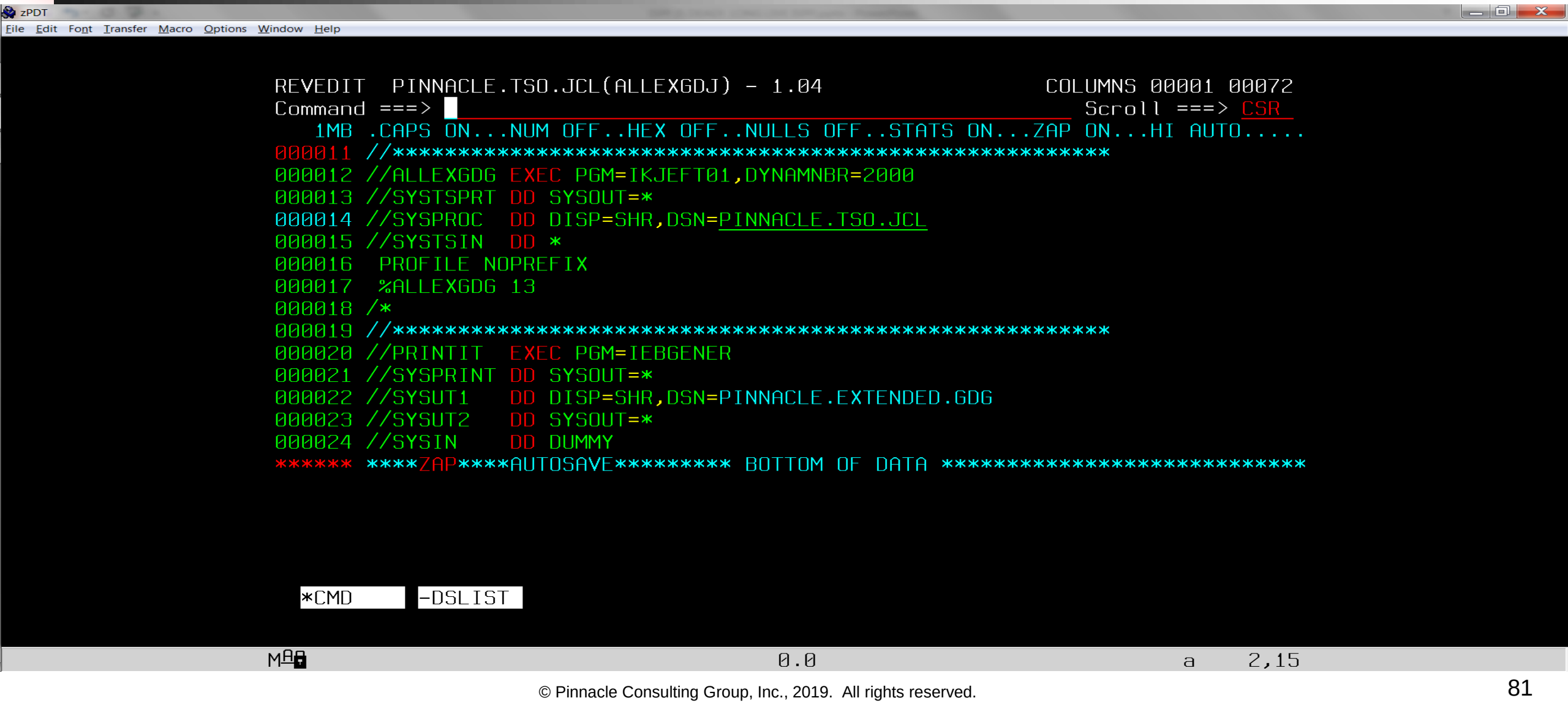
REVEDIT

- Ran some tests with FB80 file
- ISPF can edit 13M lines (~991MB), goes to browse with 14M lines
- REVEDIT can edit 20M lines (~1.5GB), goes to browse with 21M lines
  - ~50% more than ISPF can handle
- REVEDIT supports most ISPF Edit commands and adds function
- UNDO/REDO/RECOVER unavailable if LRECL > 4084 or recs > 65535
- Records updated this edit session have line numbers shown in TURQ
- Updated lines can be located via L UPD (NEXT/PREV/FIRST/LAST)
- DSNCHECK ON will highlight (un)cataloged data sets in JCL

REVEDIT



REVEDIT



PDS86

- PDS86 is still the best piece of z/OS freeware out there
- File 182 at www.cbttape.org
- Grab my presentation PDS - The Swiss Army Knife of Utilities
- Powerful utility to operate on groups of members by many criteria
  - Userid, Last Modified Date, Created, Size, etc.
  - Just about any field associated with the member can be selection criteria
- Maintained by John Kalinich, with assistance from ISPF Cabal members
- Following are slides showing new function added in the last 12 months

PDS86 - New Features

- PGLITE subset of PDSEGEN, provides Edit/View/Browse, some others
- MAP: Binder API now used for program objects and load modules
  - Enhanced reporting
  - Thank you Greg Price
- RSECT reporting (R-O)
- RMODE64 section reporting
- Private code section reporting ($PRVnnnn)
- Binder class reporting (deferred load is labeled)
- ATTRIB/DIRENTRY/VERIFY: RMODE64 reporting

PDS86 - New Features (cont'd)

- The following messages were added:
  - PDS879E RESIDENCE MODE 64 AND ADDRESSING MODE xxx ARE INCOMPATIBLE
  - PDS198I COPY, RENAME, AND REPRO DO NOT PRESERVE PDSE V2 MEMBER GENERATIONS
  - PDS983E MAXGENS VALUE IS > 2,000,000,000
  - PDS984E MAXGENS VALUE EXCEEDS THE SMS LIMIT
  - PDS199I LARGEST FREE AREA ON VOLUME volser IS nnnnnn CYLINDERS
- COPY: MAXGENS parameter added for NEW DSNTYPE(LIBRARY) output dataset allocations.

PDS86 - New Features (cont'd)

- PDSVERS: Save PDS version display setting in ISPF Profile
- O.UT: Added new options to P86OPUX (primary cmd utility)
  - FLIP Switch between current and previous data sets
  - CHEAT PDS cheat sheet for long subcommands
  - VERS Display PDS version number in Starboard area
  - YALE Compare two PDS files with Yale COMPARE command
  - PGLITE A simpler version, or subset, of PDSEGEN
- PTSEXP: Member name of ALL will expand all members of the input PDS to an output PDS
- HISTORY: Display Enterprise COBOL compiler options when they are program objects in a PDSE

PDS86 - New Features (cont'd)

- PDS200I: Display number of extents for PDSE libraries
- SHAXMIT: SHA1SUM, SHA2SUM, SHA224S, SHA3SUM, and SHA5SUM in TSO XMIT format (load modules)
- O.UT: Added new options to P86LOUX (line cmd utility)
  - RVIV RESTORE deleted versions of member
  - RNS RENAME SWAP for PDSE members
- SMPJOBS: SMP jobs to install the PDS command package. JCL, IDCAMS, and the UCLIN you need for a SMP install.
- IF: PACKED/NOPACKED parameters were added
- OFFLOAD/LOAD/PTSEXP: NEW, TRK/CYL, SPACE, and VOLUME parameters for output file allocation were added

PDS86 - New Features (cont'd)

- FIND/IF: NEWML added to THEN/ELSE actions to reset MEMLIST
- CONTROL: TOGGLES displays the state of switches set by CONTROL
- CONTROL: MEMLIST toggles whether RESET should be added to all MEMLIST subcommands
- ATTRIB: PDS856E STOW ERROR, R15=28, R0=00000000 when STOW replace attempted for a PDSE load library with the LKEDDATE parameter specified, fixed by Greg Price
- PDSQINST: Quick install exec
- $TSO8*: 8-character TSO userid/prefix support for z/OS 2.3
- PDSHELP: Add doc for functions CAX, LISTA, LISTC, LISTV, SET, TCT

PDS86 - New Features (cont'd)

- PDS859E COMMAND IKJEFTSR WAS NOT FOUND
  - LPD search in PDSMAIN failed for LPA modules because of LPD entry size increase (40 to 48) in z/OS 2.3, LPD search replaced with LOAD macro
- O.UT: Added new option to P86LLUX (line cmd utility for load)
  - CS Display CSECTs for load module
- O.UT: Added new options to P86OPUX (primary cmd utility)
  - FP FInd ISPF packed members
  - MODEL Allocate a new data set based on the active data set
  - COPMSG Extract XMIT/IEBCOPY summary messages from SYSOUT
  - CHGM CHANGE dataset menu
  - IRL Last 30 Referenced DSNAMEs

Finally…

- If you have any questions, please let me know
- If you have an idea for enhancing ISPF, please let me know
- If you want to try your hand at coding for CUCI, please let me know
- JOIN THE REVOLUTION!
- Drop me an Email pinncons@rochester.rr.com