

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)

pincons@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

```
File Edit Font Transfer Macro Options Window Help
zPDT
------( PDSE V2 Member Generations 5.5.3 )---- Row 1 to 9 of 9
Command ==> _____ Scroll ==> CSR
DSN=PINNACLE.PDSEV2.TEST - MaxGens=3
  Name      Gen  Abs  Created      Changed      VV.MM      Size      Mod  ID
- AMEM       0    0  2019/08/01  2019/08/01  20:40  01.00      1      0  IBMUSER
- AMEM      -1    2  2019/08/01  2019/08/01  20:40  01.01      2      0  IBMUSER
/ BMEM       0    0  2019/08/01  2019/08/01  20:43  01.00      1      0  IBMUSER
| BMEM      -1   49  2019/08/01  2019/08/01  20:43  01.00      1      0  IBMUSER
  BMEM      -2   47  2019/08/01  2019/08/01  20:43  01.00      1      0  IBMUSER
- LINES10M   0    0  2019/08/01  2019/08/01  01:08  01.00     10M      0  IBMUSER
- LINES11M   0    0  2019/08/01  2019/08/01  01:10  01.00     11M      0  IBMUSER
- LINES13M   0    0  2019/08/01  2019/08/01  01:16  01.00     13M      0  IBMUSER
- LINES20M   0    0  2017/05/25  2017/05/25  14:43  01.00     20M      0  IBMUSER
***** Pages: 442,857      Percent Util: 90 *****
*PDSEGEN
MAB 0.3 a 9,2
```

PDSEGEN Line Commands

```
File Edit Font Transfer Macro Options Window Help zPDT
```

```
------( PDSE V2 Member Generations 5.5.3 )---- Row 1 to 9 of 9
Command ==> Scroll ==> CSR

DSN=PINNACLE.PDSEV2.TEST - MaxGens=3

Na
AM
AM
/ BM
BM
BM
LI
LI
LI
LI
****
```

```
------( PDSE V2 Generations Select 5.5.3 )-----
Command ==> _____

Line selection processing for member:  BMEM

Generation 0 (base) selection options.

Enter a valid member selection option:  █

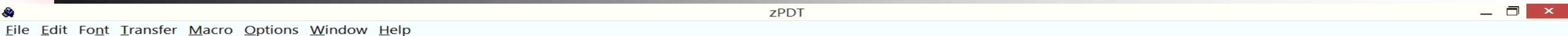
  A Attrib      E Edit      Q RenSwap    V View
  B Browse      I Info      R Rename      X eXecute
  C Copy        J Submit JCL  T Tryit
  D Delete      K Clone      U User
```

```
Mod ID
0 IBMUSER
0 IBMUSER
0 IBMUSER
0 IBMUSER
0 IBMUSER
0 IBMUSER
0 IBMUSER
0 IBMUSER
0 IBMUSER
0 IBMUSER
*****
```

```
*PDSEGEN
```

```
MB 0.0 a 13,48
```

PDSEGEN Line Commands



----- (PDSE V2 Member Generations 5.5.3)----- Row 1 to 9 of 9
Command ==> _____ Scroll ==> CSR

DSN=PINNACLE.PDSEV2.TEST - MaxGens=3

Name	Gen	Abs	Created	Changed	VV.MM	Size	Mod	ID
AMEM	0	0	2019/08/01	2019/08/01 20:40	01.00	1	0	IBMUSER
AMEM	-1	2	2019/08/01	2019/08/01 20:40	01.01	2	0	IBMUSER
BMEM	0	0	2019/08/01	2019/08/01 20:43	01.00	1	0	IBMUSER
BMEM	-1	49	2019/08/01	2019/08/01 20:43	01.00	1	0	IBMUSER
BMEM	-2	47	2019/08/01	2019/08/01 20:43	01.00	1	0	IBMUSER
LINES10M	0	0	2019/08/01	2019/08/01 01:08	01.00	10M	0	IBMUSER
LINES11M	0	0	2019/08/01	2019/08/01 01:10	01.00	11M	0	IBMUSER
LINES13M	0	0	2019/08/01	2019/08/01 01:16	01.00	13M	0	IBMUSER
LINES20M	0	0	2017/05/25	2017/05/25 14:43	01.00	20M	0	IBMUSER

***** Pages: 442,857 Percent Util: 90 *****

*PDSEGEN

PDSEGEN Line Commands

```
File Edit Font Transfer Macro Options Window Help zPDT
```

```
------( PDSE V2 Member Generations 5.5.3 )---- Row 1 to 9 of 9
Command ==> Scroll ==> CSR

DSN=PINNACLE.PDSEV2.TEST - MaxGens=3

Na
AM -----( PDSE V2 Generations Select 5.5.3 )-----
AM Command ==> _____
BM
/ BM Line selection processing for member: BMEM
BM generation: -1
LI
LI Non-0 Generation selection options.
LI
LI Enter a valid member selection option: █
****

      B Browse      I Info      T Tryit      X eXecute
      D Delete      J Submit    U User      Z Compare
      G Recover      P Promote    V View

Mod ID
0 IBMUSER
0 IBMUSER
0 IBMUSER
0 IBMUSER
0 IBMUSER
0 IBMUSER
0 IBMUSER
0 IBMUSER
*****

*PDSEGEN
```

```
MA 0.0 a 14,48
```



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](http://www.cbttape.org) 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

```
zPDT
File Edit Font Transfer Macro Options Window Help

Menu Utilities Help

Display ISPCMDS Row 10 to 24 of 92
Command ==> Scroll ==> CSR

The command table is currently open, it cannot be modified. Use the view(V)
row command to see an entire entry.

Verb T Action
-----
CRETRIEV 0 CRETRIEV
CUAATTR 0 SELECT PGM(ISPOPT) PARM(ISPOPT11) SCRNAME(SETTINGS)
CURSOR 0 CURSOR
DDLST 0 SELECT PGM(ISRDDN) NEWAPPL(ISR) SUSPEND SCRNAME(DDLST) PARM
DOWN 0 SETVERB
DSLST 0 SELECT PGM(ISRDSLST) PARM(DSL &ZPARM) SUSPEND SCRNAME(DSLST)
DTEST 0 SELECT PGM(ISPYDTST) PARM(&ZPARM) NOFUNC SCRNAME(DTEST)
END 0 SETVERB
ENVIRON 0 SELECT PGM(ISPENV) PARM(&ZPARM) SCRNAME(SETTINGS)
EPDF 0 SELECT CMD(%ISREPDF &ZPARM) NEWAPPL(ISR) SUSPEND
EXHELP 0 SELECT PGM(ISPTUTOR) PARM(&ZPARM) NOFUNC SCRNAME(HELP)
EXIT 0 EXIT
EXPAND 0 SETVERB
FKA 0 SELECT PGM(ISPOPF) PARM(FKA,&ZPARM) NOFUNC SUSPEND
HELP 0 SELECT PGM(ISPTUTOR) PARM(&ZPARM) NOFUNC SCRNAME(HELP)
```

MA 0.0 a 4,15

Updated ISPCMDS

```
zPDT
File Edit Font Transfer Macro Options Window Help

Menu Utilities Help

Display ISPCMDS Row 10 to 24 of 92
Command ==> Scroll ==> CSR

The command table is currently open, it cannot be modified. Use the view(V)
row command to see an entire entry.

Verb T Action
-----
CRETRIEV 4 CRETRIEV
CUAATTR 0 SELECT PGM(ISPOPT) PARM(ISPOPT11) SCRNAME(SETTINGS)
CURSOR 0 CURSOR
DDLIST 3 SELECT PGM(ISRDDN) NEWAPPL(ISR) SUSPEND SCRNAME(DDLIST) PARM
DOWN 0 SETVERB
DSLIST 3 SELECT PGM(ISRDSLST) PARM(DSL &ZPARM) SUSPEND SCRNAME(DSLIST
DTEST 0 SELECT PGM(ISPYDTST) PARM(&ZPARM) NOFUNC SCRNAME(DTEST)
END 0 SETVERB
ENVIRON 3 SELECT PGM(ISPENV) PARM(&ZPARM) SCRNAME(SETTINGS)
EPDF 0 SELECT CMD(%ISREPDF &ZPARM) NEWAPPL(ISR) SUSPEND
EXHELP 0 SELECT PGM(ISPTUTOR) PARM(&ZPARM) NOFUNC SCRNAME(HELP)
EXIT 0 EXIT
EXPAND 3 SETVERB
FKA 0 SELECT PGM(ISPOPF) PARM(FKA,&ZPARM) NOFUNC SUSPEND
HELP 0 SELECT PGM(ISPTUTOR) PARM(&ZPARM) NOFUNC SCRNAME(HELP)

MAB 0.0 a 4,15
```




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

```
zPDT
File Edit Font Transfer Macro Options Window Help

Menu Options View Utilities Compilers Help

DSLIST - Data Sets Matching PINNACLE.ISPF.DSN*          Row 1 of 198
Command ==>  Scroll ==> CSR

Command - Enter "/" to select action          Message          Volume
-----
//d      PINNACLE.ISPF.DSN00001                  *VSAM*
          PINNACLE.ISPF.DSN00001.DATA             USER00
          PINNACLE.ISPF.DSN00001.INDEX             USER00
          PINNACLE.ISPF.DSN00002                  *VSAM*
          PINNACLE.ISPF.DSN00002.DATA             USER00
          PINNACLE.ISPF.DSN00002.INDEX             USER00
          PINNACLE.ISPF.DSN00003                  *VSAM*
          PINNACLE.ISPF.DSN00003.DATA             USER00
          PINNACLE.ISPF.DSN00003.INDEX             USER00
          PINNACLE.ISPF.DSN00004                  *VSAM*
          PINNACLE.ISPF.DSN00004.DATA             USER00
          PINNACLE.ISPF.DSN00004.INDEX             USER00
          PINNACLE.ISPF.DSN00005                  *VSAM*
          PINNACLE.ISPF.DSN00005.DATA             USER00
          PINNACLE.ISPF.DSN00005.INDEX             USER00
          PINNACLE.ISPF.DSN00006                  *VSAM*
          PINNACLE.ISPF.DSN00006.DATA             USER00
          PINNACLE.ISPF.DSN00006.INDEX             USER00
          PINNACLE.ISPF.DSN00007                  USER11
          PINNACLE.ISPF.DSN00008                  USER07
          PINNACLE.ISPF.DSN00009                  USER02
          PINNACLE.ISPF.DSN00010                  *VSAM*
          PINNACLE.ISPF.DSN00010.DATA             USER00
          PINNACLE.ISPF.DSN00010.INDEX             USER00
//      PINNACLE.ISPF.DSN00011                  *VSAM*
```

Block DELETES for VSAM in 3.4 Without Stopping

```
zPDT
File Edit Font Transfer Macro Options Window Help

Confirm TSO Delete

Command ==> _____

Data Set Name . : PINNACLE.ISPF.DSN00001
Volume . . . . : *VSAM*
Creation date . :

CAUTION:
If TSO delete command was issued against an uncataloged data set, a
cataloged version on a volume other than the one listed here may be
deleted.

Command . . . : USRDEL34 'PINNACLE.ISPF.DSN00001'

Enter "/" to select option
/ ☒ Set data set delete confirmation off

Instructions:
Press ENTER to confirm delete.
(The data set will be deleted.)

Press CANCEL or EXIT to cancel delete.
```

Block DELETES for VSAM in 3.4 Without Stopping

```
zPDT
File Edit Font Transfer Macro Options Window Help

Menu Options View Utilities Compilers Help

DSLIST - Data Sets Matching PINNACLE.ISPF.DSN*                               Row 1 of 198
Command ==> _____ Scroll ==> CSR

Command - Enter "/" to select action                                     Message                                     Volume
-----
PINNACLE.ISPF.DSN00001          USRDEL34 RC=0          *VSAM*
PINNACLE.ISPF.DSN00001.DATA     USRDEL34 RC=4          USER00
PINNACLE.ISPF.DSN00001.INDEX    USRDEL34 RC=4          USER00
PINNACLE.ISPF.DSN00002          USRDEL34 RC=0          *VSAM*
PINNACLE.ISPF.DSN00002.DATA     USRDEL34 RC=4          USER00
PINNACLE.ISPF.DSN00002.INDEX    USRDEL34 RC=4          USER00
PINNACLE.ISPF.DSN00003          USRDEL34 RC=0          *VSAM*
PINNACLE.ISPF.DSN00003.DATA     USRDEL34 RC=4          USER00
PINNACLE.ISPF.DSN00003.INDEX    USRDEL34 RC=4          USER00
PINNACLE.ISPF.DSN00004          USRDEL34 RC=0          *VSAM*
PINNACLE.ISPF.DSN00004.DATA     USRDEL34 RC=4          USER00
PINNACLE.ISPF.DSN00004.INDEX    USRDEL34 RC=4          USER00
PINNACLE.ISPF.DSN00005          USRDEL34 RC=0          *VSAM*
PINNACLE.ISPF.DSN00005.DATA     USRDEL34 RC=4          USER00
PINNACLE.ISPF.DSN00005.INDEX    USRDEL34 RC=4          USER00
PINNACLE.ISPF.DSN00006          USRDEL34 RC=0          *VSAM*
PINNACLE.ISPF.DSN00006.DATA     USRDEL34 RC=4          USER00
PINNACLE.ISPF.DSN00006.INDEX    USRDEL34 RC=4          USER00
PINNACLE.ISPF.DSN00007          USRDEL34 RC=0          USER11
PINNACLE.ISPF.DSN00008          USRDEL34 RC=0          USER07
PINNACLE.ISPF.DSN00009          USRDEL34 RC=0          USERE2
PINNACLE.ISPF.DSN00010          USRDEL34 RC=0          *VSAM*
PINNACLE.ISPF.DSN00010.DATA     USRDEL34 RC=4          USER00
PINNACLE.ISPF.DSN00010.INDEX    USRDEL34 RC=4          USER00
PINNACLE.ISPF.DSN00011          USRDEL34 RC=0          *VSAM*
```



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
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'
    zusruuid = 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



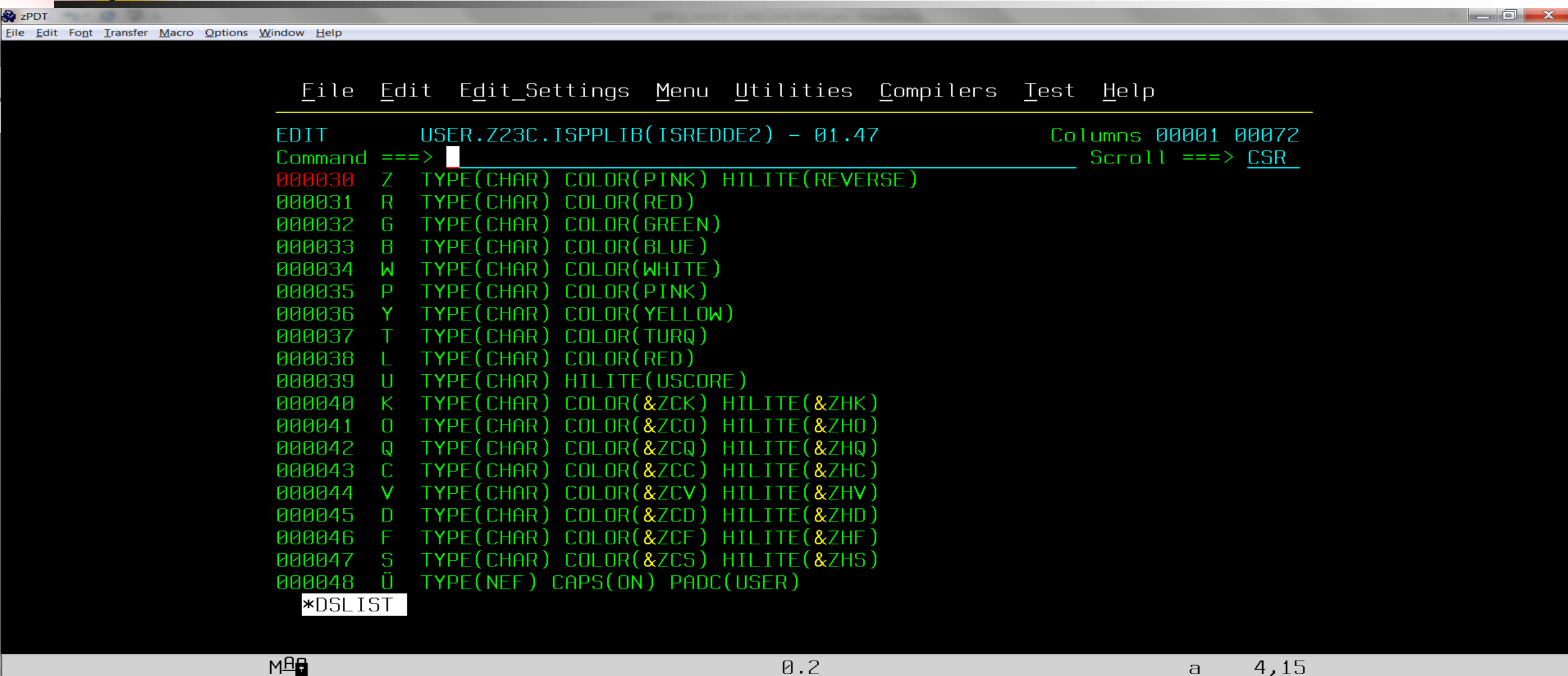
EDIT Highlighting for Java, SQL, etc!

- 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!



The screenshot shows the zPDT EDIT editor window. The title bar reads "zPDT". The menu bar includes "File", "Edit", "Font", "Transfer", "Macro", "Options", "Window", and "Help". Below the menu bar, the text "EDIT USER.Z23C.ISPPLIB(ISREDDE2) - 01.47" is displayed. To the right, it says "Columns 00001 00072" and "Scroll ==> CSR". The main editing area contains a list of commands, each with a line number, a command letter, and a description. The descriptions are color-coded: "COLOR(PINK)" is pink, "COLOR(RED)" is red, "COLOR(GREEN)" is green, "COLOR(BLUE)" is blue, "COLOR(WHITE)" is white, "COLOR(PINK)" is pink, "COLOR(YELLOW)" is yellow, "COLOR(TURQ)" is turquoise, "COLOR(RED)" is red, "HILITE(USCORE)" is underlined, and the rest are in green. The list ends with "TYPE(NEF) CAPS(ON) PADC(USER)". Below the list, the text "*DSL IST" is visible. The status bar at the bottom shows "MBA", "0.2", "a", and "4,15".

```
EDIT USER.Z23C.ISPPLIB(ISREDDE2) - 01.47 Columns 00001 00072
Command ==> Scroll ==> CSR
000030 Z TYPE(CHAR) COLOR(PINK) HILITE(REVERSE)
000031 R TYPE(CHAR) COLOR(RED)
000032 G TYPE(CHAR) COLOR(GREEN)
000033 B TYPE(CHAR) COLOR(BLUE)
000034 W TYPE(CHAR) COLOR(WHITE)
000035 P TYPE(CHAR) COLOR(PINK)
000036 Y TYPE(CHAR) COLOR(YELLOW)
000037 T TYPE(CHAR) COLOR(TURQ)
000038 L TYPE(CHAR) COLOR(RED)
000039 U TYPE(CHAR) HILITE(USCORE)
000040 K TYPE(CHAR) COLOR(&ZCK) HILITE(&ZHK)
000041 O TYPE(CHAR) COLOR(&ZCO) HILITE(&ZHO)
000042 Q TYPE(CHAR) COLOR(&ZCQ) HILITE(&ZHQ)
000043 C TYPE(CHAR) COLOR(&ZCC) HILITE(&ZHC)
000044 V TYPE(CHAR) COLOR(&ZCV) HILITE(&ZHV)
000045 D TYPE(CHAR) COLOR(&ZCD) HILITE(&ZHD)
000046 F TYPE(CHAR) COLOR(&ZCF) HILITE(&ZHF)
000047 S TYPE(CHAR) COLOR(&ZCS) HILITE(&ZHS)
000048 Ü TYPE(NEF) CAPS(ON) PADC(USER)
*DSL IST
```



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 HIGHLIGHT command

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

```
HIGHLIGHT language | OFF
```

```
language = CARLA | FORTRAN | JAVA | PYTHON | RUBY | SHELL | SQL
```

OFF turns off extended language highlighting.

HIGHLIGHT BRACE toggles brace highlighting on and off.

HIGHLIGHT BRACKET toggles bracket highlighting on and off.

HIGHLIGHT DISPLAY | DISP shows current state of HIGHLIGHT options.

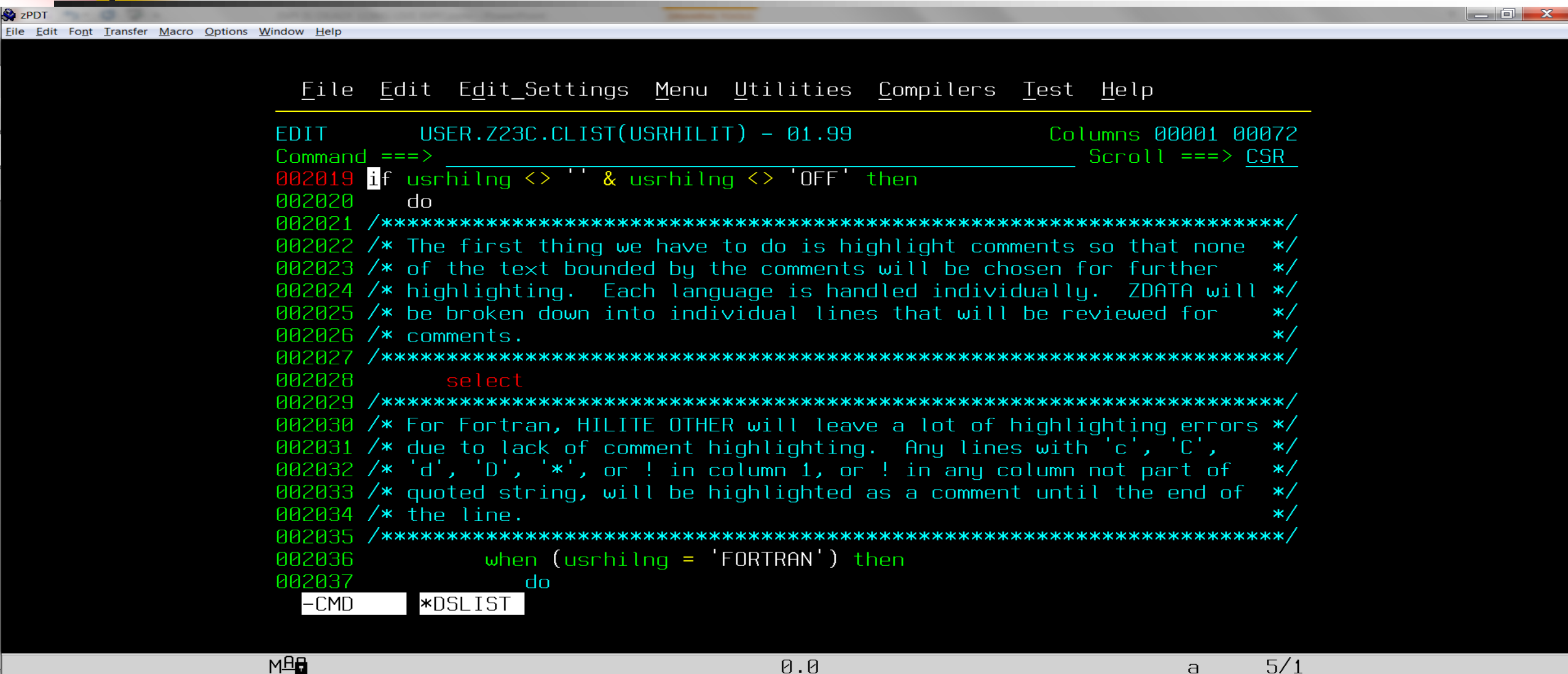
HIGHLIGHT HELP | ? displays this HELP text.

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



The image shows a screenshot of the zPDT (z/OS Personal Data Transfer) software interface. The window title is 'zPDT'. The menu bar includes 'File', 'Edit', 'Font', 'Transfer', 'Macro', 'Options', 'Window', and 'Help'. Below the menu bar, there is a secondary menu with 'File', 'Edit', 'Edit_Settings', 'Menu', 'Utilities', 'Compilers', 'Test', and 'Help'. The main editing area displays Fortran code with line numbers 002019 through 002037. The code includes a conditional statement 'if usrhilng <> '' & usrhilng <> 'OFF' then' followed by a 'do' block. Inside the 'do' block, there are several lines of comments explaining the highlighting process for Fortran, mentioning 'HILITE OTHER' and 'HILITE OTHER will leave a lot of highlighting errors'. The code ends with a 'when (usrhilng = 'FORTRAN') then' block. At the bottom of the editing area, there are two buttons: '-CMD' and '*DSLIST'. The status bar at the bottom of the window shows 'MMA', '0.0', 'a', and '5/1'.

```
EDIT          USER.Z23C.CLIST(USRHILIT) - 01.99          Columns 00001 00072
Command ==>          Scroll ==> CSR
002019 if usrhilng <> '' & usrhilng <> 'OFF' then
002020     do
002021 /******
002022 /* The first thing we have to do is highlight comments so that none */
002023 /* of the text bounded by the comments will be chosen for further   */
002024 /* highlighting. Each language is handled individually. ZDATA will  */
002025 /* be broken down into individual lines that will be reviewed for    */
002026 /* comments.                                                           */
002027 /******
002028     select
002029 /******
002030 /* For Fortran, HILITE OTHER will leave a lot of highlighting errors */
002031 /* due to lack of comment highlighting. Any lines with 'c', 'C',    */
002032 /* 'd', 'D', '*', or ! in column 1, or ! in any column not part of   */
002033 /* quoted string, will be highlighted as a comment until the end of  */
002034 /* the line.                                                           */
002035 /******
002036         when (usrhilng = 'FORTRAN') then
002037     do
-CMD      *DSLIST
```

Converting Special Chars to Blanks, Uppercase

```
zPDT
File Edit Font Transfer Macro Options Window Help

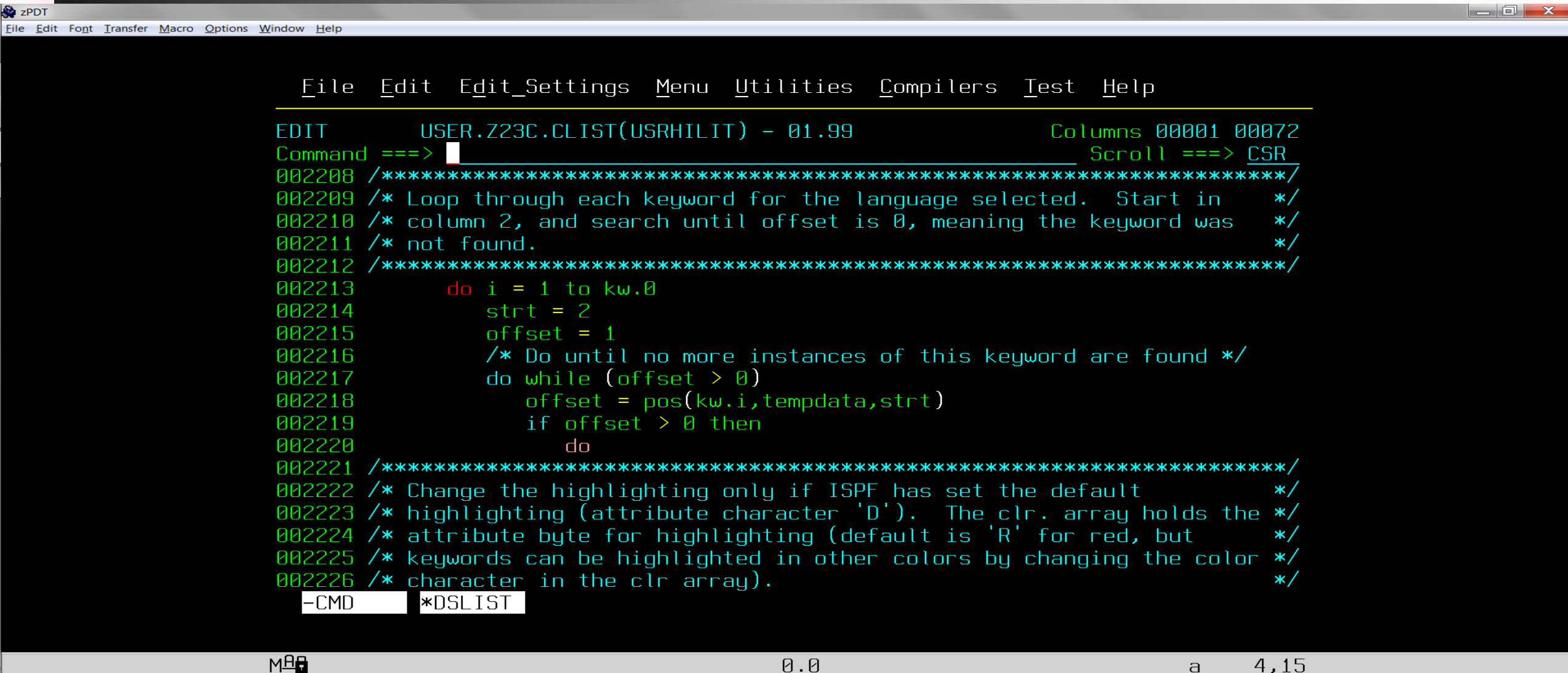
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT      USER.Z23C.CLIST(USRHILIT) - 01.99      Columns 00001 00072
Command ==>  Scroll ==> CSR

002196      tempdata = zdata
002197 /******
002198 /* The double commas are necessary to make the translate command */
002199 /* work properly. The first comma is the comma for the function, */
002200 /* the second comma will continue the line. */
002201 /******
002202      tempdata = translate(tempdata,,
002203                          '152A2B2F141B1C260102030405060708090A0B0C0D1316171D20'x)
002204      tempdata = translate(tempdata,
002205                          '{ } [ ] ( ) # + - * / = < > & ^ | : % , ' )
002206      parse upper var tempdata tempdata
002207 /******
002208 /* Loop through each keyword for the language selected. Start in */
002209 /* column 2, and search until offset is 0, meaning the keyword was */
002210 /* not found. */
002211 /******
002212      do i = 1 to kw.0
002213          strt = 2
002214      -CMD      *DSL IST

MBA 0.0 a 4,15
```

Highlighting Keywords



The image shows a screenshot of the zPDT (z/OS Performance Diagnostic Tool) interface. The window title is 'zPDT'. The menu bar includes 'File', 'Edit', 'Font', 'Transfer', 'Macro', 'Options', 'Window', and 'Help'. Below the menu bar, there is a toolbar with icons for file operations. The main editing area displays a code editor with the following content:

```
File Edit Edit_Settings Menu Utilities Compilers Test Help

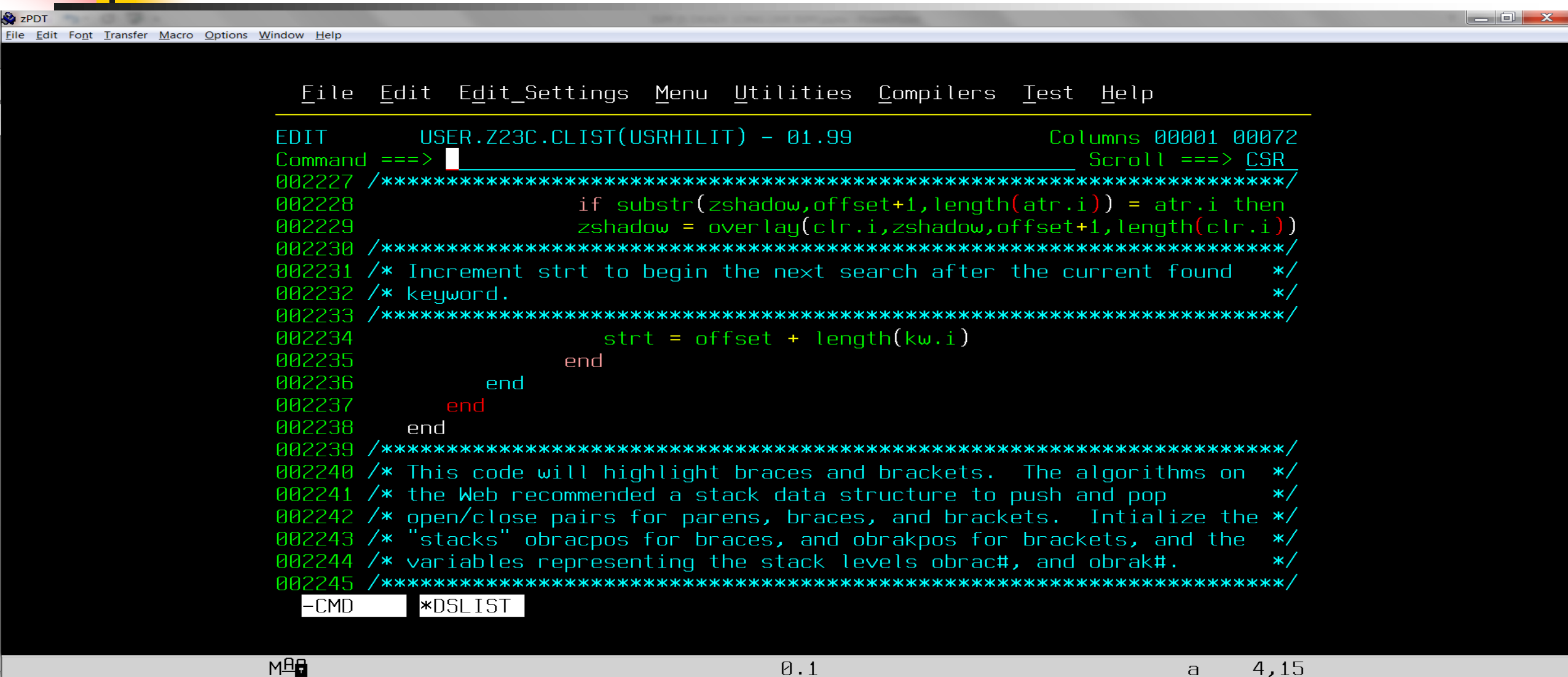
EDIT          USER.Z23C.CLIST(USRHILIT) - 01.99          Columns 00001 00072
Command ==>  Scroll ==> CSR

002208 /*****
002209 /* Loop through each keyword for the language selected. Start in */
002210 /* column 2, and search until offset is 0, meaning the keyword was */
002211 /* not found. */
002212 /*****
002213         do i = 1 to kw.0
002214             strt = 2
002215             offset = 1
002216             /* Do until no more instances of this keyword are found */
002217             do while (offset > 0)
002218                 offset = pos(kw.i,tempdata,strt)
002219                 if offset > 0 then
002220                     do
002221 /*****
002222 /* Change the highlighting only if ISPF has set the default */
002223 /* highlighting (attribute character 'D'). The clr. array holds the */
002224 /* attribute byte for highlighting (default is 'R' for red, but */
002225 /* keywords can be highlighted in other colors by changing the color */
002226 /* character in the clr array). */
-CMD  *DSLIST
```

At the bottom of the window, there is a status bar with the following information:

MPB 0.0 a 4,15

Highlighting Keywords

A screenshot of a zPDT window with a dark background. The window title is 'zPDT' and the menu bar includes 'File', 'Edit', 'Font', 'Transfer', 'Macro', 'Options', 'Window', and 'Help'. Below the menu bar is a secondary menu with 'File', 'Edit', 'Edit_Settings', 'Menu', 'Utilities', 'Compilers', 'Test', and 'Help'. The main text area displays code with syntax highlighting: comments are in cyan, keywords like 'if', 'end', and 'end' are in green, and variable names like 'zshadow', 'clr.i', 'strt', 'kw.i', 'obracpos', and 'obrakpos' are in red. Line numbers 002227 through 002245 are on the left. At the bottom, there are two input fields: '-CMD' and '*DSLIST'. The status bar at the very bottom shows 'MBA', '0.1', 'a', and '4,15'.

```
zPDT
File Edit Font Transfer Macro Options Window Help
File Edit Edit_Settings Menu Utilities Compilers Test Help

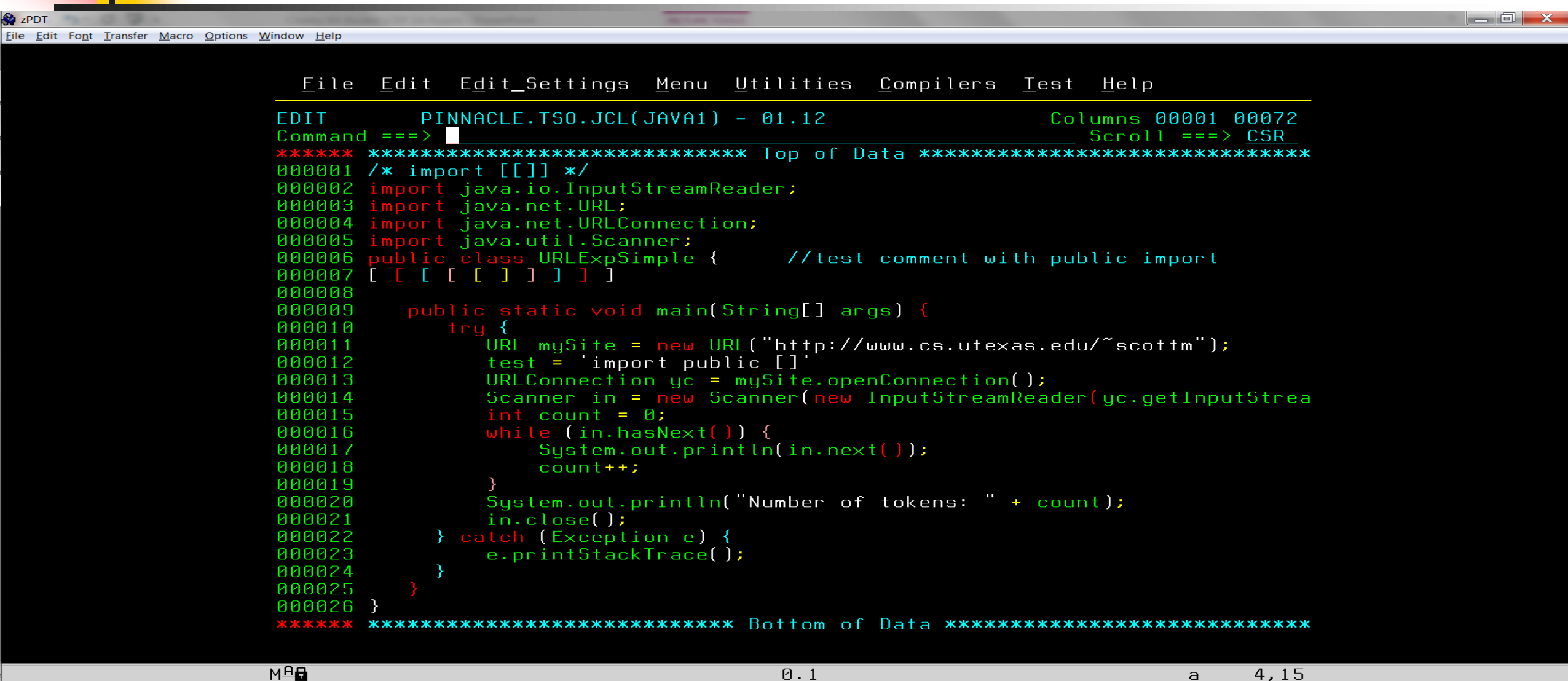
EDIT          USER.Z23C.CLIST(USRHILIT) - 01.99          Columns 00001 00072
Command ==>  Scroll ==> CSR

002227 /*****
002228         if substr(zshadow,offset+1,length(atr.i)) = atr.i then
002229         zshadow = overlay(clr.i,zshadow,offset+1,length(clr.i))
002230 /*****
002231 /* Increment strt to begin the next search after the current found */
002232 /* keyword. */
002233 /*****
002234         strt = offset + length(kw.i)
002235         end
002236     end
002237 end
002238 end
002239 /*****
002240 /* This code will highlight braces and brackets. The algorithms on */
002241 /* the Web recommended a stack data structure to push and pop */
002242 /* open/close pairs for parens, braces, and brackets. Intialize the */
002243 /* "stacks" obracpos for braces, and obrakpos for brackets, and the */
002244 /* variables representing the stack levels obrac#, and obrak#. */
002245 /*****

-CMD  *DSLIST 

MBA 0.1 a 4,15
```

HIGHLIGHT JAVA, BRACKET, and BRACE

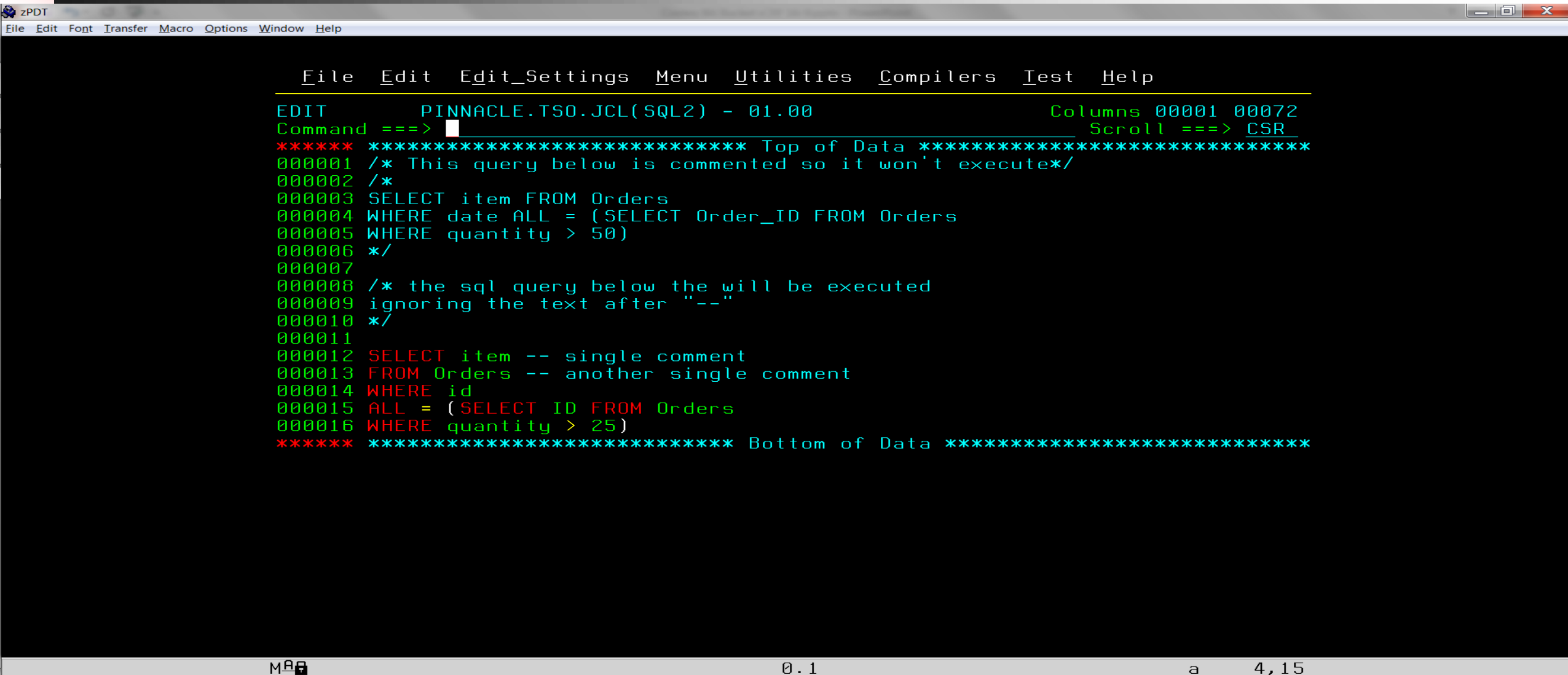


The image shows a screenshot of the zPDT (z/OS Performance Diagnostic Tool) editor window. The window title is "zPDT". The menu bar includes "File", "Edit", "Font", "Transfer", "Macro", "Options", "Window", and "Help". The status bar at the bottom shows "MBA", "0.1", "a", and "4,15".

The editor displays a Java file named "PINNACLE.TSO.JCL(JAVA1) - 01.12". The code is as follows:

```
EDIT          PINNACLE.TSO.JCL(JAVA1) - 01.12          Columns 00001 00072
Command ==>                                Scroll ==> CSR
***** Top of Data *****
000001  /* import [[]] */
000002  import java.io.InputStreamReader;
000003  import java.net.URL;
000004  import java.net.URLConnection;
000005  import java.util.Scanner;
000006  public class URLExpSimple {           //test comment with public import
000007  [ [ [ [ [ ] ] ] ] ]
000008
000009      public static void main(String[] args) {
000010          try {
000011              URL mySite = new URL("http://www.cs.utexas.edu/~scottm");
000012              test = 'import public []';
000013              URLConnection yc = mySite.openConnection();
000014              Scanner in = new Scanner(new InputStreamReader(yc.getInputStream()));
000015              int count = 0;
000016              while (in.hasNext()) {
000017                  System.out.println(in.next());
000018                  count++;
000019              }
000020              System.out.println("Number of tokens: " + count);
000021              in.close();
000022          } catch (Exception e) {
000023              e.printStackTrace();
000024          }
000025      }
000026  }
***** Bottom of Data *****
```

HIGHLITE SQL



The image shows a terminal window titled 'zPDT' with a menu bar containing 'File', 'Edit', 'Font', 'Transfer', 'Macro', 'Options', 'Window', and 'Help'. Below the menu bar is a toolbar with icons for file operations. The main area of the window displays SQL code with syntax highlighting. The code is as follows:

```
EDIT          PINNACLE.TSO.JCL(SQL2) - 01.00          Columns 00001 00072
Command ==>           Scroll ==> CSR
***** Top of Data *****
000001  /* This query below is commented so it won't execute*/
000002  /*
000003  SELECT item FROM Orders
000004  WHERE date ALL = (SELECT Order_ID FROM Orders
000005  WHERE quantity > 50)
000006  */
000007
000008  /* the sql query below the will be executed
000009  ignoring the text after "--"
000010  */
000011
000012  SELECT item -- single comment
000013  FROM Orders -- another single comment
000014  WHERE id
000015  ALL = (SELECT ID FROM Orders
000016  WHERE quantity > 25)
***** Bottom of Data *****
```

The status bar at the bottom of the window displays 'M', a lock icon, '0.1', 'a', and '4,15'.



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

```
zPDT
File Edit Font Transfer Macro Options Window Help

File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT      /S0W1/etc/inetd.conf      Columns 00001 00072
Command ==> create inetd1.conf      Scroll ==> CSR
***** Top of Data *****
c99999 ###
000002 # Internet server configuration database
000003 #
000004 # (C) COPYRIGHT International Business Machines Corp. 1985, 2001
000005 # All Rights Reserved
000006 # Licensed Materials - Property of IBM
000007 #
000008 # US Government Users Restricted Rights - Use, duplication or
000009 # disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
000010 #
000011 # /etc/inetd.conf
000012 #
000013 #           Internet server configuration database
000014 #
000015 # $01=PYQ0049, HOT7705, 010130, PDJP: Correct paths and remove
000016 #           unsupported services (FIN APAR 0W45915
000017 #
000018 # Services can be added and deleted by deleting or inserting a
-MOUNTTA DTEST *UDLIST

MAB 0.2 a 6/1
```

USS path for COMPARE, COPY, CREATE, REPLACE

```
zPDT
File Edit Font Transfer Macro Options Window Help

File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT      /S0W1/etc/inetd.conf      File created
Command ==> _____ Scroll ==> CSR

***** Top of Data *****
000001 ###
000002 # Internet server configuration database
000003 #
000004 # (C) COPYRIGHT International Business Machines Corp. 1985, 2001
000005 # All Rights Reserved
000006 # Licensed Materials - Property of IBM
000007 #
000008 # US Government Users Restricted Rights - Use, duplication or
000009 # disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
000010 #
000011 # /etc/inetd.conf
000012 #
000013 #           Internet server configuration database
000014 #
000015 # $01=PYQ0049, HOT7705, 010130, PDJP: Correct paths and remove
000016 #           unsupported services (FIN APAR 0W45915
000017 #
000018 # Services can be added and deleted by deleting or inserting a
-MOUNTTA DTEST *UDLIST

MBA 0.1 a 7,9
```

USS path for COMPARE, COPY, CREATE, REPLACE

```
zPDT
File Edit Font Transfer Macro Options Window Help

Menu Utilities View Options Help

z/OS UNIX Directory List Row 1 to 14 of 45
Command ==> Scroll ==> CSR

Pathname . : /S0W1/etc
EUID . . . : 0
Command Filename Message Type Perm Owner
-----
. Dir 755 OMVSKERN
.. Dir 755 OMVSKERN
csh.cshrc File 644 OMVSKERN
csh.login File 644 OMVSKERN
dbb Dir 755 OMVSKERN
dfs Dir 755 OMVSKERN
hosts File 755 OMVSKERN
httpd.conf File 700 OMVSKERN
httpd.envvars File 600 OMVSKERN
ics_pics.conf File 600 OMVSKERN
inetd.conf File 000 OMVSKERN
inetd1.conf File 000 OMVSKERN
init.options File 000 OMVSKERN
ioepdcf Syml 777 OMVSKERN
-MOUNTTA DTEST *UDLIST
```



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  010D010101040202030403060408040A0B1305090A0A060B0C13070C0D13080D0E13...
ZMTMARR = MTBFCOLS 01080000012301010223020203040303050404040608050507030606040A0707080A
ZMTFARR = MTBMCOLS 01080000022301010123020203040303050404040608050507030606040A0707080A
```

- 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

```
zPDT
File Edit Font Transfer Macro Options Window Help

File Edit Edit_Settings Menu Utilities Compilers Test Help

ISREDDE2 PINNACLE.ISPF.KEYWORD(USRDEFLT) - 01.16 Columns 00001 00072
Command ==>  Scroll ==> CSR
***** ***** Top of Data *****
==MSG> -CAUTION- Data contains invalid (non-display) characters. Use command
==MSG> ==> FIND P'.' to position cursor to these
000001 /*****
000002 /* This is the ISPF USR keyword file to define ISPF defaults for */
000003 /* options not included in ISPCCONF. */
000004 /* */
000005 /* Instructions */
000006 /* */
000007 /* Lines with '/' in columns 1 and 2 are comments. If a keyword is */
000008 /* commented out, the ISPF default value is provided for reference. */
000009 /* */
000010 /* Keyword lines are in the format: */
000011 /* */
000012 /* keyword = value[,1] */
000013 /* */
000014 /* where value is the keyword value, and ',1' indicates that even if */
000015 /* the user has another value specified, FORCE the value specified */
000016 /* here into the user profile. Lining up the equal signs is */
*DSLST
```

Set ISPF Defaults for Options not in ISPCCONF

```
zPDT
File Edit Font Transfer Macro Options Window Help

File Edit Edit_Settings Menu Utilities Compilers Test Help

ISREDDE2 PINNACLE.ISPF.KEYWORD(USRDEFLT) - 01.15 Columns 00001 00072
Command ==> Scroll ==> CSR
000019 /*****
000020 /* PFKeys */
000021 /*****
000022 /*PF1 = HELP */
000023 /*PF2 = SPLIT */
000024 /*PF3 = END */
000025 /*PF4 = RETURN */
000026 /*PF5 = RFIND */
000027 /*PF6 = RCHANGE */
000028 /*PF7 = UP */
000029 /*PF8 = DOWN */
000030 /*PF9 = SWAP */
000031 /*PF10 = LEFT */
000032 /*PF11 = RIGHT */
000033 /*PF12 = RETRIEVE */
000034 PF13 = :ts
000035 /*PF14 = SPLIT */
000036 /*PF15 = END */
000037 PF16 = :tf
*DSLIST
```

Set ISPF Defaults for Options not in ISPCCONF

```
zPDT
File Edit Font Transfer Macro Options Window Help

File Edit Edit_Settings Menu Utilities Compilers Test Help

ISREDDE2 PINNACLE.ISPF.KEYWORD(USRDEFLT) - 01.15 Columns 00001 00072
Command ==> Scroll ==> CSR
000046 /*****
000047 /* MEMLIST SRCHFOR Options */
000048 /* */
000049 /* The SRCHFOR_LISTING_DSN supports the use of system symbols. */
000050 /* Support was also added for a pseudo-symbol &SYSUID, which will */
000051 /* be replaced with the ISPF userid for the user. */
000052 /*****
000053 MEMLIST_SRCHFOR_LISTING_DSN = '&SYSUID..#&SYSNAME..SFMLLIST'
000054 /*MEMLIST_SRCHFOR_MIXED_MODE = / */
000055 /*MEMLIST_SRCHFOR_ANY_CASE = / */
000056 /*MEMLIST_SRCHFOR_ASCII = / */
000057 /*MEMLIST_SRCHFOR_VIEW_OUTPUT = / */
000058 /*MEMLIST_SRCHFOR_SAVE_OUTPUT = / */
000059 MEMLIST_SRCHFOR_SET_EDIT_FIND_STRING = /
000060 MEMLIST_SRCHFOR_SET_BROWSE_FIND_STRING = /
000061 MEMLIST_SRCHFOR_FILTER_LIST = /
000062 /*****
000063 /* DSLIST SRCHFOR Options */
000064 /* */
*DSLST
```

Set ISPF Defaults for Options not in ISPCCONF

```
zPDT
File Edit Font Transfer Macro Options Window Help

File Edit Edit_Settings Menu Utilities Compilers Test Help

ISREDDE2 PINNACLE.ISPF.KEYWORD(USRDEFLT) - 01.15 Columns 00001 00072
Command ==> Scroll ==> CSR
000098 /*****
000099 /*                      Calendar Options                      */
000100 /*                      */
000101 /* Start Date:  1 - Sunday (default), 2 - Monday, 3 - Saturday */
000102 /*                      */
000103 /* Colors are single character values in hex.                  */
000104 /*                      */
000105 /* Colors      :  x'31' - White, x'32' - Red,  x'33' - Blue,    */
000106 /*              x'34' - Green, x'35' - Pink, x'36' - Yellow,    */
000107 /*              x'37' - Turq,  x'40' - CUA Default              */
000108 /*                      */
000109 /* Time Format:  1 - 12-hour (default), 2 - 24-hour             */
000110 /*****
000111 CALENDAR_START_DATE                      = 1
000112 CALENDAR_SCROLL_BUTTON_COLOR            =
000113 CALENDAR_HEADING_DATE_COLOR             =
000114 CALENDAR_HEADING_TEXT_COLOR             =
000115 CALENDAR_WEEKDAY_COLOR                  =
000116 CALENDAR_SATURDAY/SUNDAY_COLOR         =
*DSLIST
```

Set ISPF Defaults for Options not in ISPCCONF

```
zPDT
File Edit Font Transfer Macro Options Window Help

File Edit Edit_Settings Menu Utilities Compilers Test Help

ISREDDE2 USER.Z23C.ISPPLIB(TOM@PRIM) - 01.43 Columns 00001 00072
Command ==> Scroll ==> CSR
000227 /*****
000228 /* This code will get the USRKWNEW variable for the Rexx exec and */
000229 /* set it to 'YES' if it doesn't exist, which will create the */
000230 /* keyword profile variables. */
000231 /*****
000232 VGET (USRKWNEW) PROFILE
000233 IF (&USRKWNEW = &Z)
000234     &USRKWNEW = 'YES'
000235 &USRFORCE = 'NO'
000236 /*****
000237 /* This panel Rexx code will process and set ISPF keyword options */
000238 /* not covered in ISPCCONF. */
000239 /*****
000240 &USRNMLST = 'USRKWNEW USRFORCE +
000241             ZPF01 ZPF02 ZPF03 ZPF04 ZPF05 ZPF06 +
000242             ZPF07 ZPF08 ZPF09 ZPF10 ZPF11 ZPF12 +
000243             ZPF13 ZPF14 ZPF15 ZPF16 ZPF17 ZPF18 +
000244             ZPF19 ZPF20 ZPF21 ZPF22 ZPF23 ZPF24 +
000245             ZMSFMIXV ZMSFANYC ZMSFASCI ZMSFVIEW +
000246             *DSLST
```

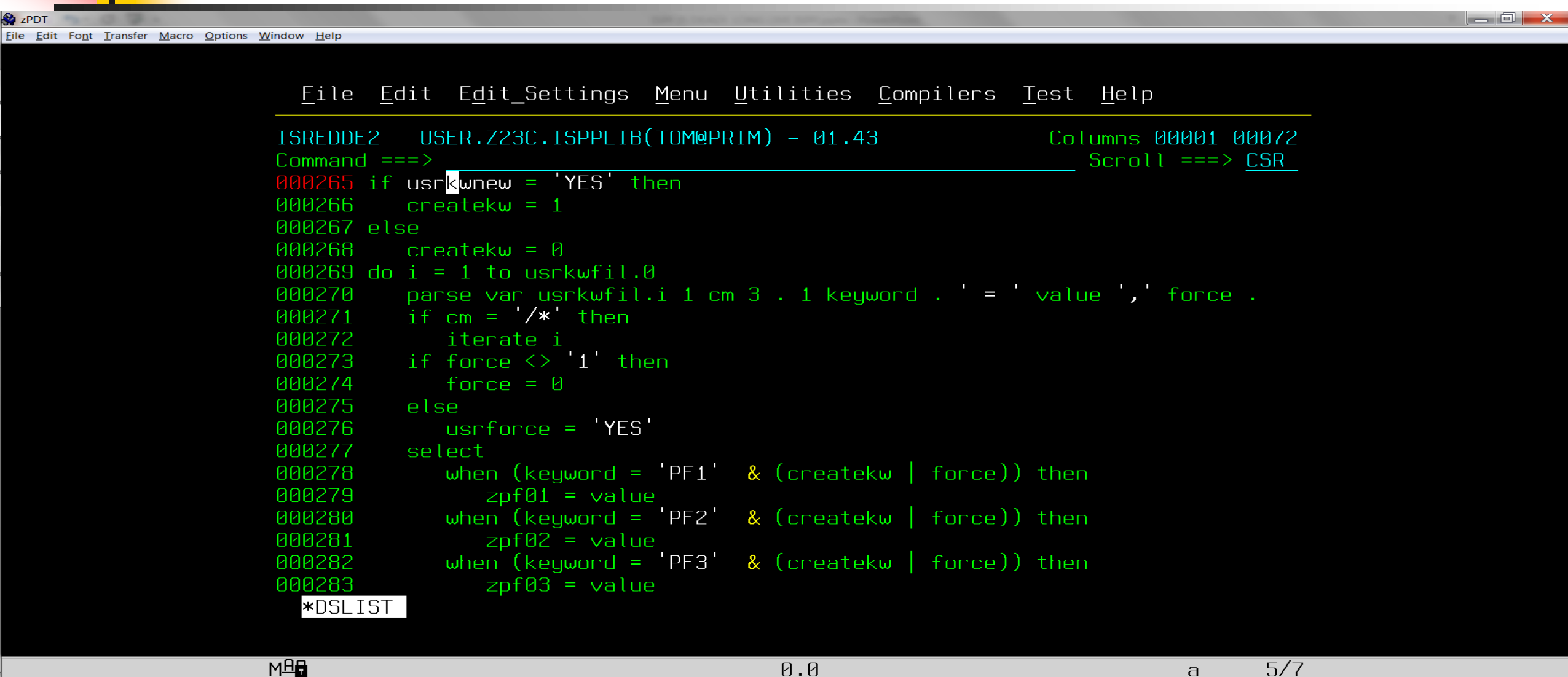
Set ISPF Defaults for Options not in ISPCCONF

```
zPDT
File Edit Font Transfer Macro Options Window Help

File Edit Edit_Settings Menu Utilities Compilers Test Help
ISREDDE2 USER.Z23C.ISPPLIB(TOM@PRIM) - 01.43 Columns 00001 00072
Command ==> Scroll ==> CSR
000246 ZMSFSAVE ZMSFFILT ZMSFEDTF ZMSFBROF +
000247 ZUDFMIXV ZUDFANYC ZUDFASCI ZUDFVIEW +
000248 ZUDFSAVE ZUDFLIST ZUDFEDTF ZUDFBROF +
000249 ZMSFLIST ZCALSTRT ZCALGCSB ZCALGCHD +
000250 ZCALGCHT ZCALGCWD ZCALGCWE ZCALGCTD +
000251 ZCALTIME ZUS4LIST ZUS4MIXV ZUS4MIGR +
000252 ZUS4XCLO ZUS4VIEW ZUS4ANYC ZUS4XNF +
000253 ZUS4ASCI ZUS4TOTS ZUS4STOP +
000254 ZUS4XCLO ZUS4VIEW ZUS4ANYC ZUS4XNF +
000255 ZUS4ASCI ZUS4TOTS ZUS4STOP +
000256 ZULWFNMC ZULFSUBC ZULSHCTM +
000257 ZULFCO NV ZULC MDOM ZULDCO NV ZULBYEDV +
000258 ZULDPMOV ZULUXPCV ZULBYEDF ZULDPMOF +
000259 ZULUXPCF'
000260 *REXX(&USRNMLST)
000261 address tso "ALLOC FI(USRKWFIL) SHR",
000262 "DA('PINNACLE.ISPF.KEYWORD(USRDEFLT)')'"
000263 "EXECIO * DISKR USRKWFIL (OPEN STEM USRKWFIL. FINIS)"
000264 address tso "FREE FI(USRKWFIL)"
*DSLST

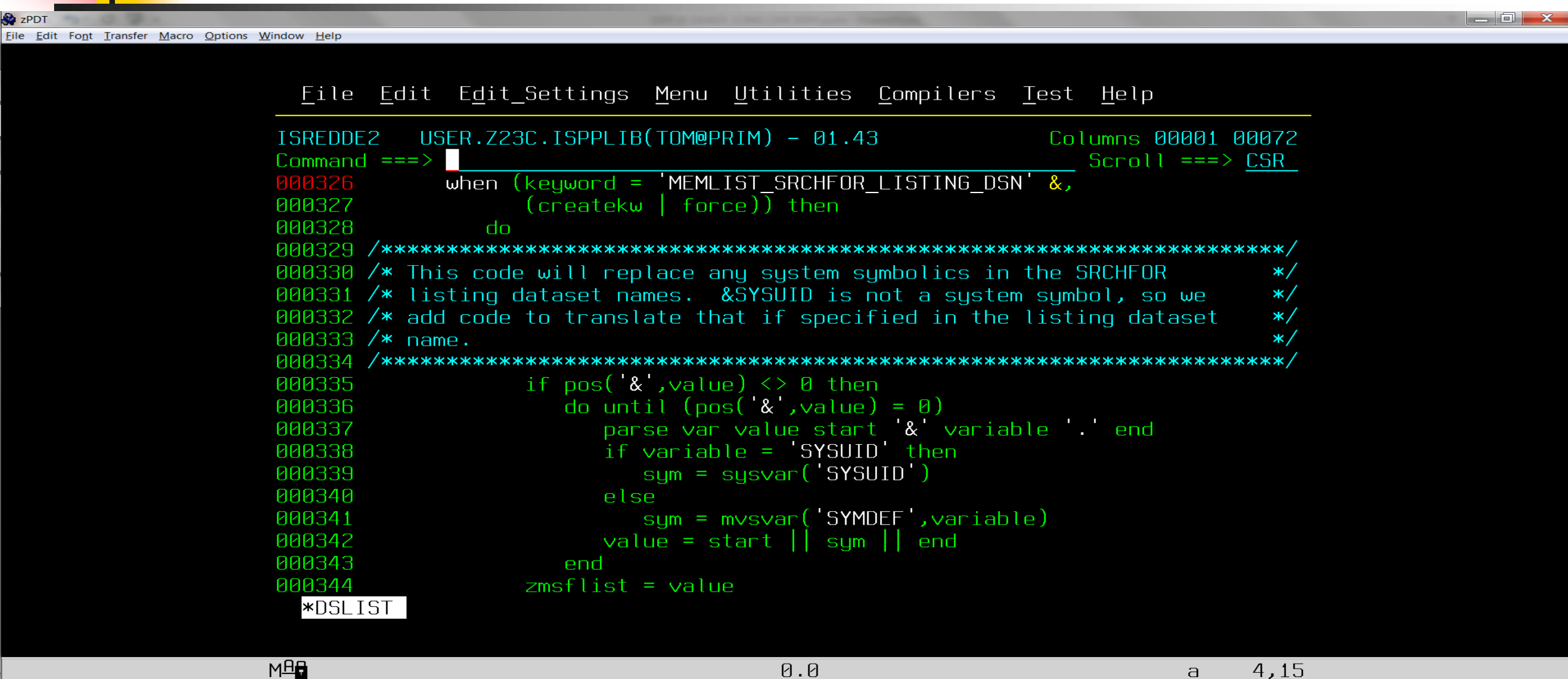
MBA 0.0 a 4,15
```


Set ISPF Defaults for Options not in ISPCCONF

A screenshot of a zPDT window displaying ISPF defaults configuration. The window has a menu bar with 'File', 'Edit', 'Font', 'Transfer', 'Macro', 'Options', 'Window', and 'Help'. Below the menu bar, there is a status line showing 'ISREDDE2 USER.Z23C.ISPPLIB(TOM@PRIM) - 01.43' and 'Columns 00001 00072'. The main area contains a command stream starting with 'Command ==>' followed by a series of lines for setting defaults. The command stream includes an 'if' statement for 'usrkwnew' and a 'select' statement for 'keyword'. The command stream ends with '*DSL IST'. The status bar at the bottom shows 'MBA', '0.0', 'a', and '5/7'.

```
File Edit Font Transfer Macro Options Window Help
ISREDDE2 USER.Z23C.ISPPLIB(TOM@PRIM) - 01.43 Columns 00001 00072
Command ==> Scroll ==> CSR
000265 if usrkwnew = 'YES' then
000266     createkw = 1
000267 else
000268     createkw = 0
000269 do i = 1 to usrkwfil.0
000270     parse var usrkwfil.i 1 cm 3 . 1 keyword . ' = ' value ',' force .
000271     if cm = '/*' then
000272         iterate i
000273     if force <> '1' then
000274         force = 0
000275     else
000276         usrforce = 'YES'
000277     select
000278         when (keyword = 'PF1' & (createkw | force)) then
000279             zpf01 = value
000280         when (keyword = 'PF2' & (createkw | force)) then
000281             zpf02 = value
000282         when (keyword = 'PF3' & (createkw | force)) then
000283             zpf03 = value
*DSL IST
```

Set ISPF Defaults for Options not in ISPCCONF

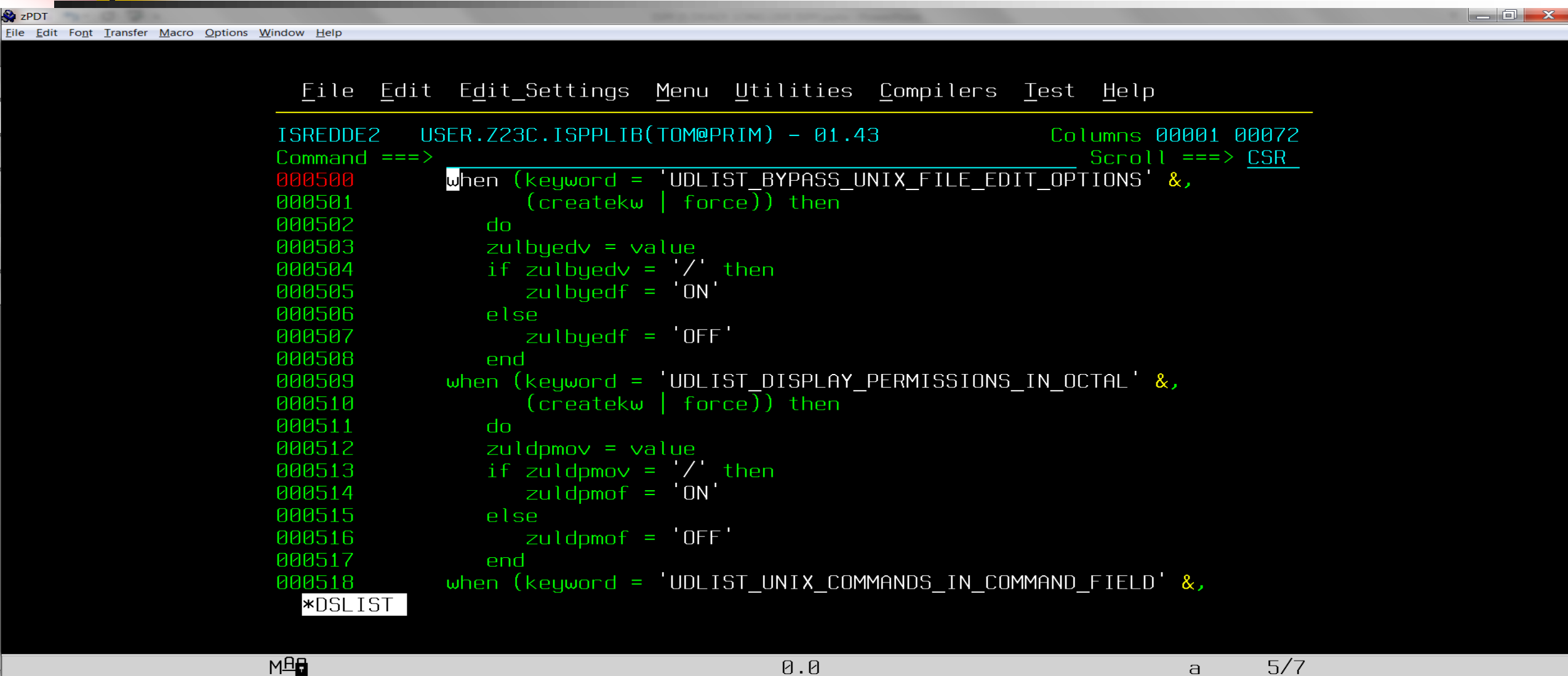
A screenshot of a zPDT window displaying an ISPF command stream. The window has a menu bar with 'File', 'Edit', 'Font', 'Transfer', 'Macro', 'Options', 'Window', and 'Help'. Below the menu bar, the command stream is shown with line numbers 000326 through 000344. The stream includes a 'when' statement for keyword 'MEMLIST_SRCHFOR_LISTING_DSN' and a 'do' block containing code to replace system symbolics. The status bar at the bottom shows 'MBA', '0.0', 'a', and '4,15'.

```
zPDT
File Edit Font Transfer Macro Options Window Help

File Edit Edit_Settings Menu Utilities Compilers Test Help

ISREDDE2 USER.Z23C.ISPPLIB(TOM@PRIM) - 01.43 Columns 00001 00072
Command ==> Scroll ==> CSR
000326 when (keyword = 'MEMLIST_SRCHFOR_LISTING_DSN' &,
000327         (createkw | force)) then
000328     do
000329 /******
000330 /* This code will replace any system symbolics in the SRCHFOR */
000331 /* listing dataset names. &SYSUID is not a system symbol, so we */
000332 /* add code to translate that if specified in the listing dataset */
000333 /* name. */
000334 /******
000335         if pos('&',value) <> 0 then
000336             do until (pos('&',value) = 0)
000337                 parse var value start '&' variable '.' end
000338                 if variable = 'SYSUID' then
000339                     sym = sysvar('SYSUID')
000340                 else
000341                     sym = mvsvvar('SYMDEF',variable)
000342                 value = start || sym || end
000343             end
000344         zmsflist = value
*DSLST
```

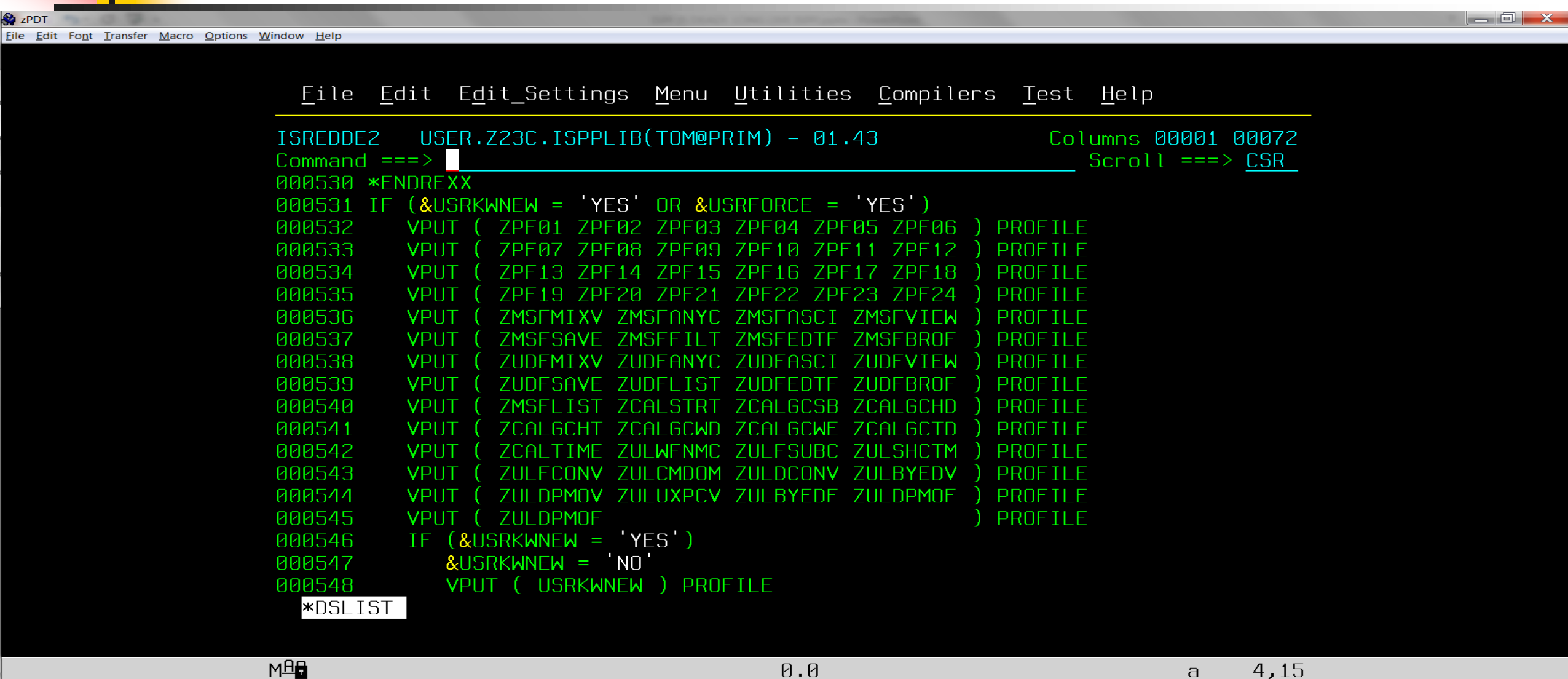
Set ISPF Defaults for Options not in ISPCCONF

A screenshot of a zPDT window displaying an ISPF configuration screen. The window has a menu bar with 'File', 'Edit', 'Font', 'Transfer', 'Macro', 'Options', 'Window', and 'Help'. Below the menu bar, the text 'ISREDDE2 USER.Z23C.ISPPLIB(TOM@PRIM) - 01.43' is shown on the left, and 'Columns 00001 00072' is on the right. The main area contains a configuration script with green text on a black background. The script starts with 'Command ==>' followed by a series of 'when' and 'do' blocks. The first block sets 'UDLIST_BYPASS_UNIX_FILE_EDIT_OPTIONS' to 'ON' if the keyword is present. The second block sets 'UDLIST_DISPLAY_PERMISSIONS_IN_OCTAL' to 'ON' if the keyword is present. The third block sets 'UDLIST_UNIX_COMMANDS_IN_COMMAND_FIELD' to 'ON' if the keyword is present. The script ends with '*DSLIST'. The status bar at the bottom shows 'MBA', '0.0', 'a', and '5/7'.

```
File Edit Font Transfer Macro Options Window Help

ISREDDE2 USER.Z23C.ISPPLIB(TOM@PRIM) - 01.43 Columns 00001 00072
Command ==> Scroll ==> CSR
000500 when (keyword = 'UDLIST_BYPASS_UNIX_FILE_EDIT_OPTIONS' &,
000501         (createkw | force)) then
000502     do
000503         zulbyedv = value
000504         if zulbyedv = '/' then
000505             zulbyedf = 'ON'
000506         else
000507             zulbyedf = 'OFF'
000508         end
000509     when (keyword = 'UDLIST_DISPLAY_PERMISSIONS_IN_OCTAL' &,
000510         (createkw | force)) then
000511         do
000512             zuldpmov = value
000513             if zuldpmov = '/' then
000514                 zuldpmof = 'ON'
000515             else
000516                 zuldpmof = 'OFF'
000517             end
000518     when (keyword = 'UDLIST_UNIX_COMMANDS_IN_COMMAND_FIELD' &,
000519         (createkw | force)) then
000520         do
000521             zulcmdf = value
000522             if zulcmdf = '/' then
000523                 zulcmdf = 'ON'
000524             else
000525                 zulcmdf = 'OFF'
000526             end
000527         end
000528     end
000529 *DSLIST
```

Set ISPF Defaults for Options not in ISPCCONF

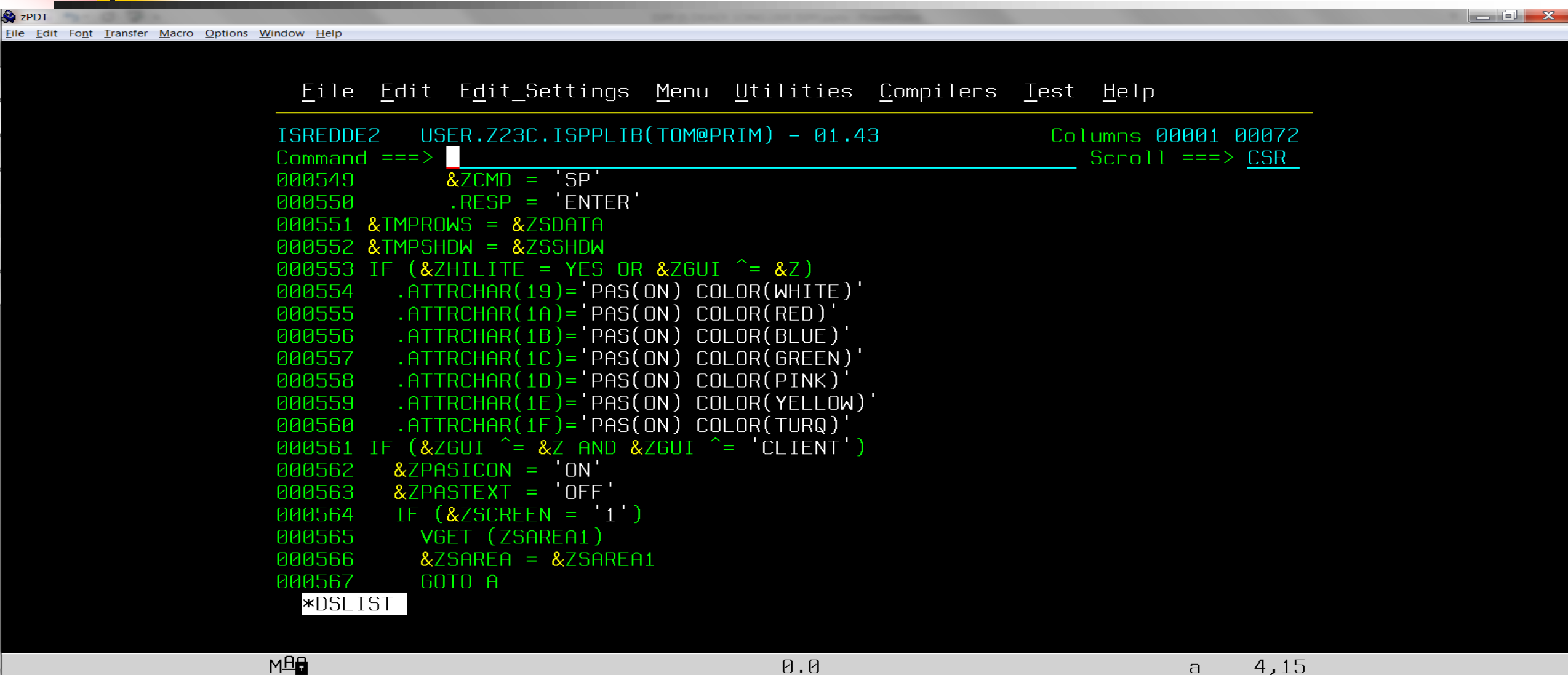


The image shows a screenshot of a zPDT (z/OS Personal Data Transfer) window. The window has a menu bar with options: File, Edit, Font, Transfer, Macro, Options, Window, and Help. Below the menu bar, there is a status bar showing 'ISREDDE2 USER.Z23C.ISPPLIB(TOM@PRIM) - 01.43' and 'Columns 00001 00072'. The main area displays a command stream in green text on a black background. The commands are as follows:

```
Command ==> 
000530 *ENDREXX
000531 IF (&USRKNEW = 'YES' OR &USRFORCE = 'YES')
000532     VPUT ( ZPF01 ZPF02 ZPF03 ZPF04 ZPF05 ZPF06 ) PROFILE
000533     VPUT ( ZPF07 ZPF08 ZPF09 ZPF10 ZPF11 ZPF12 ) PROFILE
000534     VPUT ( ZPF13 ZPF14 ZPF15 ZPF16 ZPF17 ZPF18 ) PROFILE
000535     VPUT ( ZPF19 ZPF20 ZPF21 ZPF22 ZPF23 ZPF24 ) PROFILE
000536     VPUT ( ZMSFMIXV ZMSFANYC ZMSFASCI ZMSFVIEW ) PROFILE
000537     VPUT ( ZMSFSAVE ZMSFFILT ZMSFEDTF ZMSFBROF ) PROFILE
000538     VPUT ( ZUDFMIXV ZUDFANYC ZUDFASCI ZUDFVIEW ) PROFILE
000539     VPUT ( ZUDFSAVE ZUDFLIST ZUDFEDTF ZUDFBROF ) PROFILE
000540     VPUT ( ZMSFLIST ZCALSTRT ZCALGCSB ZCALGCHD ) PROFILE
000541     VPUT ( ZCALGCHT ZCALGCWD ZCALGCWE ZCALGCTD ) PROFILE
000542     VPUT ( ZCALTIME ZULWFNMC ZULFSUBC ZULSHCTM ) PROFILE
000543     VPUT ( ZULFCONV ZULCMDOM ZULDCONV ZULBYEDV ) PROFILE
000544     VPUT ( ZULDPMOV ZULUXPCV ZULBYEDF ZULDPMOF ) PROFILE
000545     VPUT ( ZULDPMOF ) PROFILE
000546     IF (&USRKNEW = 'YES')
000547         &USRKNEW = 'NO'
000548     VPUT ( USRKNEW ) PROFILE
*DSL
```

The status bar at the bottom shows 'M00' on the left, '0.0' in the center, and 'a 4,15' on the right.

Set ISPF Defaults for Options not in ISPCCONF

A screenshot of a zPDT window displaying ISPF configuration code. The window has a menu bar with 'File', 'Edit', 'Font', 'Transfer', 'Macro', 'Options', 'Window', and 'Help'. Below the menu bar, the text 'ISREDDE2 USER.Z23C.ISPPLIB(TOM@PRIM) - 01.43' is shown on the left, and 'Columns 00001 00072' and 'Scroll ==> CSR' are on the right. The main area contains a series of commands and conditional statements in green text on a black background. The commands include setting attributes for the command line, response, and various display options. The code ends with a GOTO statement and a label *DSLST. The status bar at the bottom shows 'MBA', '0.0', 'a', and '4,15'.



REVEDIT

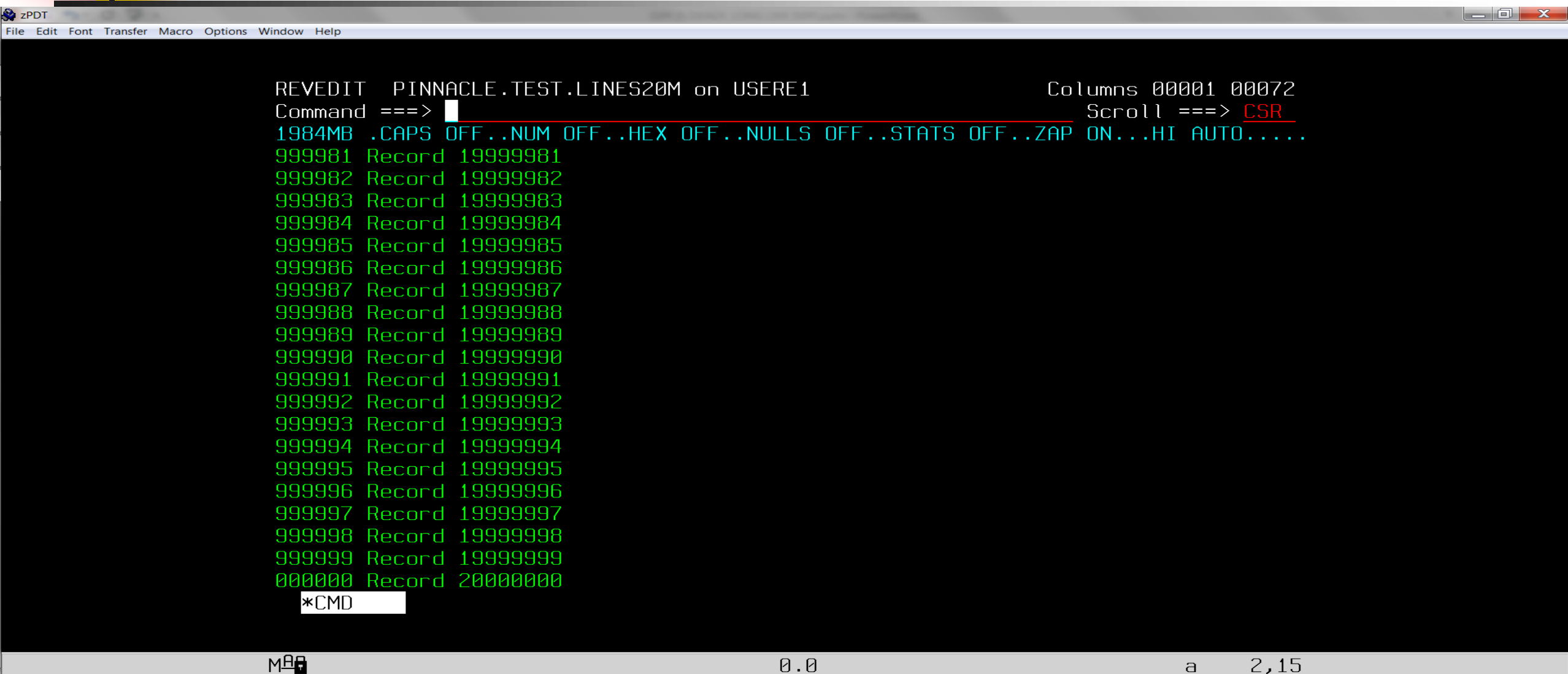
- REVEDIT is part of Greg Price's REVIEW package on CBT Mods Tape
- Grab Greg's FILE134 [here](#), and FILE135 [here](#)
- REVEDIT partially satisfies RFE 112742 for editing larger files
- Uses SCOPE-SINGLE extendable dataspace
 - Dataspace limited to 2GB or smaller by installation settings
 - Cut/Paste 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

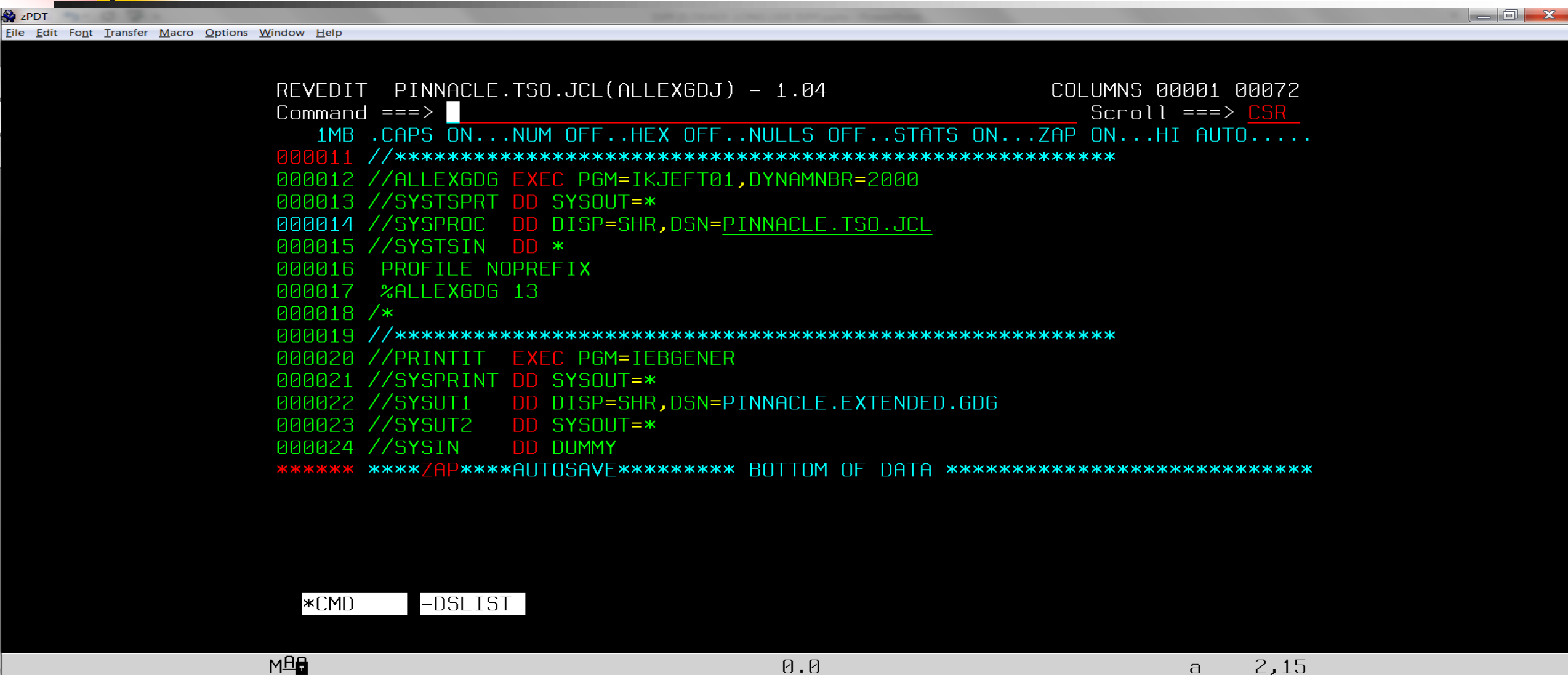
A screenshot of a zPDT window titled 'zPDT'. The window has a menu bar with 'File', 'Edit', 'Font', 'Transfer', 'Macro', 'Options', 'Window', and 'Help'. The main area is a black terminal window with white and green text. The text shows the REEDIT interface for a file named 'PINNACLE.TEST.LINES20M' on 'USERE1'. It displays various settings like 'Columns 00001 00072', 'Scroll ==> CSR', and a list of records from 999981 to 000000. A cursor is positioned at the end of the 'Command ==>' line. The bottom status bar shows 'MBA', '0.0', 'a', and '2,15'.

```
zPDT
File Edit Font Transfer Macro Options Window Help

REVEDIT PINNACLE.TEST.LINES20M on USERE1
Columns 00001 00072
Command ==>
1984MB .CAPS OFF..NUM OFF..HEX OFF..NULLS OFF..STATS OFF..ZAP ON...HI AUTO.....
999981 Record 19999981
999982 Record 19999982
999983 Record 19999983
999984 Record 19999984
999985 Record 19999985
999986 Record 19999986
999987 Record 19999987
999988 Record 19999988
999989 Record 19999989
999990 Record 19999990
999991 Record 19999991
999992 Record 19999992
999993 Record 19999993
999994 Record 19999994
999995 Record 19999995
999996 Record 19999996
999997 Record 19999997
999998 Record 19999998
999999 Record 19999999
000000 Record 20000000
*CMD
```

MBA 0.0 a 2,15

REVEDIT



The image shows a screenshot of a zPDT window titled "zPDT". The menu bar includes "File", "Edit", "Font", "Transfer", "Macro", "Options", "Window", and "Help". The main window displays the output of the REVEDIT command, showing the JCL for PINNACLE.TSO.JCL. The output includes various JCL statements such as //ALLEXGDG EXEC PGM=IKJEFT01, //SYSTSPRT DD SYSOUT=*, //SYSPROC DD DISP=SHR, DSN=PINNACLE.TSO.JCL, //SYSTSIN DD *, PROFILE NOPREFIX, %ALLEXGDG 13, /*, //PRINTIT EXEC PGM=IEBGENER, //SYSPRINT DD SYSOUT=*, //SYSUT1 DD DISP=SHR, DSN=PINNACLE.EXTENDED.GDG, //SYSUT2 DD SYSOUT=*, //SYSIN DD DUMMY, and a ZAP statement for AUTOSAVE. The window also shows the command line with "Command ==>" and "Scroll ==> CSR". At the bottom, there are two buttons: "*CMD" and "-DSL IST". The status bar at the bottom shows "MBA", "0.0", "a", and "2,15".

```
REVEDIT  PINNACLE.TSO.JCL(ALLEXGDJ) - 1.04          COLUMNS 00001 00072
Command ==>  Scroll ==> CSR
      1MB .CAPS ON...NUM OFF..HEX OFF..NULLS OFF..STATS ON...ZAP ON...HI AUTO.....
000011 //*****
000012 //ALLEXGDG EXEC PGM=IKJEFT01,DYNAMNBR=2000
000013 //SYSTSPRT DD SYSOUT=*
000014 //SYSPROC  DD DISP=SHR,DSN=PINNACLE.TSO.JCL
000015 //SYSTSIN  DD *
000016  PROFILE NOPREFIX
000017  %ALLEXGDG 13
000018 /*
000019 //*****
000020 //PRINTIT  EXEC PGM=IEBGENER
000021 //SYSPRINT DD SYSOUT=*
000022 //SYSUT1   DD DISP=SHR,DSN=PINNACLE.EXTENDED.GDG
000023 //SYSUT2   DD SYSOUT=*
000024 //SYSIN     DD DUMMY
*****  ****ZAP****AUTOSAVE***** BOTTOM OF DATA *****
*CMD  -DSL IST
```

MBA 0.0 a 2,15



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 nnnnnnn 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 pincons@rochester.rr.com