



Dynamic ISPF

How to Eliminate All Those Datasets in the LOGON PROC Part 2

NewEra Software - The zExchange
July 14, 2015

Thomas Conley
Pinnacle Consulting Group, Inc.
59 Applewood Drive
Rochester, NY 14612-3501
P: (585)720-0012
F: (585)723-3713
pinncons@rochester.rr.com



Abstract

Do you have a large number of LOGON procs at your site? Do you ever get tired of adding ISPF application datasets to every LOGON proc? Have you ever wondered if there's a better way to install and maintain your ISPF applications? If you answered 'YES' to any of these questions, this session is for you! Come to this popular session to learn how to invoke ISPF applications dynamically. The knowledge gained will help you to shrink your logon procs to a manageable size, eliminate redundant procs, and get better overall ISPF performance to boot! This session will include an online demonstration of Dynamic ISPF.



Agenda

- Part 1 Review
- Miscellaneous Recommendations
- Sell Dynamic ISPF to Management
- ServerPac - “The Dark Side”
- Conversion to Dynamic ISPF
- Dynamic ISPF Starter Set (DISS)
- Setting Up TSO and ISPF (WAC)
- Summary
- Finally...



Part 1 Review

- “Standard” ISPF vs. Dynamic ISPF
- Coding Rexx Driver EXECs
- ISPF Menu Construction
- Performance Considerations
- LOADLIBs, Dynamic Steplib, TSOLIB
- Solving Common Problems



Miscellaneous Recommendations

- Use naming convention for Rexx driver EXECs (e.g. use "@" prefix, @RMM invokes RMM, @IPCS invokes IPCS, etc.)
- Use security package to secure ISPF application datasets
- Don't allow LOGON PROC allocations to provide false sense of security



Miscellaneous Recommendations

- If using ISPTABL dataset, that dataset should also be first in ISPTLIB DD:

```
address ispexec "LIBDEF ISPTABL DATASET ID('SVAA.SIBTABL') STACK"  
address ispexec "LIBDEF ISPTLIB DATASET ID('SVAA.SIBTABL', "  
" 'SVAA.SIBTLIB') STACK"
```

- Use scrollable menus to eliminate multiple submenus and multi-column selection menus (ISR@PRIM is a good example)
- Panel TRANS function has 126-entry limit; OA02785 displays error message ISPP323 if limit for TRANS function exceeded

Miscellaneous Recommendations

- Following code circumvents 126-entry limit for TRANS; use to create single scrollable menu with unlimited selections:

```
) PROC
  &ZSEL = TRANS ( &ZQ
    X,EXIT
    SP, 'PGM(ISPSAM) PARM(PNS) '
    ' ', ' '
    *, '?' )
&NEXTZSEL = 'NO'
IF ( &ZSEL = ' ' )
  &NEXTZSEL = 'YES'
IF ( &ZSEL = '?' )
  &NEXTZSEL = 'YES'
IF ( &NEXTZSEL = 'YES' )
  &ZSEL = TRANS ( &ZQ
    X,EXIT
    SP, 'PGM(ISPSAM) PARM(PNS) '
    ' ', ' '
    *, '?' )
```



Miscellaneous Recommendations

- Include SCRNAME on SELECT statement in Rexx driver EXEC (instead of selection menu)
- Users with command tables or using TSO @EXEC can then take advantage of SWAP LIST processing

```
address ispexec "SELECT CMD(SIBREMM0) NOCHECK SCRNAME(SVAA) "
```


Miscellaneous Recommendations

- For ISPF applications in command table, use &ZPARM when invoking Rexx driver EXEC for nested menu selections:

```
Menu  Utilities  Help
```

```
-----  
                                Display TESTCMDS                                Row 1 to 1 of 1  
Command ===>                                Scroll ===> CSR
```

```
Insert (I), delete (D), repeat (R) and edit (E) command entries.  
Enter END command to save changes or CANCEL to end without saving.
```

Verb	T	Action
<u>ISMF</u>	<u>0</u>	SELECT CMD(%@ISMF I.&ZPARM) NOCHECK
<u>RACF</u>	<u>0</u>	SELECT CMD(%@RACF &ZPARM) NOCHECK



Miscellaneous Recommendations

- If command table applications need &ZCMD (ISMF, SMP/E, etc.), use dummy selection argument "I." before &ZPARM
- User can then enter "ISMF 1" for ISMF dataset menu, instead of "ISMF I.1"

Miscellaneous Recommendations

- Delete SYSHELP DD statement in LOGON PROC and use HELP statement in IKJTSO00 instead:

```
HELP  ENU(SYS1.HELP          /* HELP COMMAND DEFAULTS */ +
        ISP.SISPHELP        /* ISPF HELP                */ +
        )
```

- Using HELP statement in IKJTSO00 saves resources and reduces contention
- Help datasets only allocated if user enters TSO HELP command



Miscellaneous Recommendations

- Some applications do not support dynamic help from IKJTSO00, so you might have to leave SYSHELP in LOGON PROC
- CA-ACF2 does not support dynamic help, so use ACFHELP DD to satisfy ACF2 help and IKJTSO00 for everything else
- Due to problem with attention processing, OY59638 recommended using LOGON CLISTS instead of EXECs

Miscellaneous Recommendations

- By using Rexx QUEUE or PUSH command ISPF can be invoked by LOGON EXEC

```
/* Rexx */
/* trace i */
msgsave = msg()
x = msg('OFF')
address tso "ALLOC FI(ISPPROF) DA('userid()'.ISPF.ISPPROF') OLD"
if rc <> 0 then
  do
    profdsn = userid().ISPF.ISPPROF
    x = msg('ON')
    address tso "ALLOC FI(ISPPROF) DA('profdsn') NEW CATALOG",
              "CYLINDERS SPACE(1 1) DIR(45) DSORG(PO) RECFM(F B)",
              "LRECL(80) BLKSIZE(0) "
    if rc = 0 then
      say "ISPF profile dataset'"profdsn"'created"
    else
      say "Unable to allocate ISPF profile dataset'"profdsn'"
  end
x = msg(msgsave)
/* queue "TSOLIB ACT DA('tsolib dsn') " Put TSOLIB ACT here if needed */
queue "ISPF NOLOGO"
/* queue "TSOLIB DEACT" Put TSOLIB DEACT here if needed */
```



Miscellaneous Recommendations

- Rexx QUEUE command puts "ISPF NOLOGO" onto Rexx stack
- "ISPF NOLOGO" executes from READY after LOGON EXEC terminates
- OY59638 bypassed since ISPF executes in native TSO instead of Rexx environment
- Initial command on TSO LOGON panel is inhibitor to using LOGON EXEC
- That command will be executed before QUEUED or PUSHed ISPF command



Miscellaneous Recommendations

- Use LOGON EXECs instead of LOGON CLISTs wherever possible
- TSOLIB is also handled by “QUEUE”ing it before ISPF and freeing it afterward
- Thanks to Ken Burrow at Computer Sciences Corporation for sharing this Rexx QUEUE command method to enable LOGON EXECs in Rexx



Miscellaneous Recommendations

- In previous releases of z/OS, IBM has repackaged certain components into other datasets
- To handle repackaging, Rexx driver EXEC can use Rexx **mvsvar** function to interrogate **sysmvs** variable
- **sysmvs** indicates MVS release on system
- **sysmvs** can be used to make decisions in Rexx driver EXEC based on MVS release



Miscellaneous Recommendations

- In OS/390 V2R10, IBM repackaged RMM by eliminating SEDGSKL1 and SEDGTBL1
- SEDGSKL1 and SEDGTBL1 members moved to DGTSLIB and DGTTLIB, respectively
- The following @ISMF example shows how to use **mvsvar** function with **sysmvs** to allocate proper datasets for RMM support

Miscellaneous Recommendations

```
/* REXX */
/* trace i */
parse arg zcmd
.
.
/*****
/* This code determines which release of OS/390 is running because */
/* at OS/390 V2R10 and higher, RMM eliminated SEDGSKL1 and SEDGTBL1. */
/* This modification is courtesy of David Alcock at American */
/* Electric Power. */
*****/
if mvsvr('sysmvs') > 'SP6.0.9' then
  do
    address ispexec "LIBDEF ISPSLIB DATASET ID('SYS1.DGTSLIB') STACK"
    address ispexec "LIBDEF ISPTLIB DATASET ID('userid()'.ISPPROF',",
                  "'SYS1.DGTTLIB') STACK"
  end
else
  do
    address ispexec "LIBDEF ISPSLIB DATASET ID('SYS1.DGTSLIB',",
                  "'SYS1.SEDGSKL1') STACK"
    address ispexec "LIBDEF ISPTLIB DATASET ID('userid()'.ISPPROF',",
                  "'SYS1.DGTTLIB',",
                  "'SYS1.SEDGTBL1') STACK"
  end
.
.
```



Sell Dynamic ISPF to Management

- Sell Performance
 - Faster LOGON
 - Lower BLDL overhead
 - Smaller TSO working set size
 - Reduced storage and paging
- Sell Reliability and Serviceability (RAS)
 - Smaller LOGON PROC
 - Single Rexx driver EXEC
 - Immediate access to new ISPF applications
 - ENQ's not held unless ISPF dialog active



Sell Dynamic ISPF to Management

- Sell Disaster Recovery
 - Recover from disasters more quickly
 - TSO active when system and site-specific ISPF datasets restored
- Sell Security
 - Implement real security for ISPF applications
 - Secure ISPF application dataset access
 - No false sense of security provided by "Standard" ISPF LOGON proc



Server Pac - The "Dark Side"

- ServerPac implements "Standard" ISPF
 - Large LOGON PROCs
 - LOGON CLIST to free and reallocate libraries
 - ISP@MSTR where ISPF not main application
- ServerPac also includes optional function to merge similar datasets into one library
- This function most often used to merge ISPF application datasets
- Creates ISPF application datasets with (hundreds of) thousands of members



Server Pac - The "Dark Side"

- ServerPac's tacit endorsement of this ISPF install method using merged libraries is significant inhibitor to Dynamic ISPF acceptance
- LOGON PROC maintenance simplified, BUT
- BLDL overhead still significant since ISPF libraries will likely contain (hundreds of) thousands of members



Server Pac - The “Dark Side”

- Without Parallel Access Volume support, loads for ISPF dialog elements single-threaded since each merged ISPF library is on single volume
- Dynamic ISPF can spread out I/O for better performance
- Merging libraries allows all users to execute all ISPF dialogs
- Dynamic ISPF can control access to specific ISPF dialogs with dataset security



Server Pac - The “Dark Side”

- If two ISPF applications use same member name, another library must be used to resolve conflict
- If you haven't implemented merged library methodology, use Dynamic ISPF instead
- If you've implemented merged library methodology, consider Dynamic ISPF as better alternative when you install your next ServerPac



Conversion to Dynamic ISPF

- Conversion to Dynamic ISPF heavily dependent on your existing TSO/ISPF installation methodology
- Conversion accomplished for each application by:
 - Coding Dynamic ISPF Rexx driver EXEC
 - Create menu and/or command table selection
 - Thoroughly testing application's functionality
- Once applications converted, plan production cutover



Conversion to Dynamic ISPF

- Production cutover to Dynamic ISPF can be accomplished in phases, but I recommend an approach that cuts everything over at once
- This approach minimizes changes, problems, and time required to convert to Dynamic ISPF



Conversion to Dynamic ISPF

- Create LOGON PROC for building Dynamic ISPF applications, per example shown earlier
- Choose application to convert and code Rexx driver EXEC to invoke application
- Create menu and/or command table entry to invoke application
- Test selection menu/command table entry and application for functionality



Conversion to Dynamic ISPF

- Code appropriate access rules for ISPF application datasets
- If creating new LOGON PROC name(s), update each TSO user's security profile for new PROC name(s)
- Repeat these steps until each ISPF application converted and each menu selection/command table entry tested



Conversion to Dynamic ISPF

- For production cutover, create new LOGON PROCs for Dynamic ISPF
- I recommend swapping old and new PROCs via rename; using new PROC names will require updating user security profiles for access to new PROC
- Similar to LOGON PROCs, create new selection menus/command table members, and implement via rename swaps



Conversion to Dynamic ISPF

- Select small subset of users as beta test group
- I recommend systems programmers, since they usually have access to widest range of ISPF applications, and they're more likely to accept change gracefully 😊
- Communicate implementation to users
- GO FOR IT!!



Dynamic ISPF Starter Set (DISS)

- To remove inhibitor and assist conversion to Dynamic ISPF, I created a Starter Set of Rexx driver EXECs for many popular ISPF applications and made it available at:

<http://www.cbttape.org/ftp/cbt/CBT495.zip>



Dynamic ISPF Starter Set (DISS)

- DISS contains nearly 200 Rexx EXECs, panels, documentation, and other members to invoke popular ISPF applications
- If you encounter any problems with the Starter Set, or to contribute a new Rexx driver EXEC to Starter Set, please send an Email to pinncons@rochester.rr.com



Product Launch Point (PLP)

- Lionel Dyck has created a menu-driven application called Product Launch Point (PLP)
- PLP allows user to specify elements of an ISPF application (e.g. datasets, SELECT command to invoke application) and PLP will invoke ISPF application dynamically
- Check out PLP and other fine free software available from Lionel's web site:

<http://www.lbdsoftware.com/index.htm>



Dynamic STEPLIB

- Dynamic STEPLIB available for free on CBT mods tape
- MVS/JES2 Software Dynamic STEPLIB: <http://www.cbttape.org/ftp/cbt/CBT452.zip>
- Taken together, DISS, PLP, and Dynamic STEPLIB on CBTTape FILE452 are all you need to completely exploit Dynamic ISPF



Setting up TSO and ISPF (WAC)

- I'm often asked about how to set up TSO and ISPF, so here's WAC (World According to Conley) method
- Using Dynamic ISPF LOGON PROC shown earlier, create single common Dynamic ISPF LOGON PROC
- Create alias members for each functional group (e.g. SYSPROG, TSOAPPL, etc.) pointing to single common Dynamic ISPF LOGON PROC



Setting up TSO and ISPF (WAC)

- Use LOGON EXEC shown earlier in Miscellaneous Recommendations section to allocate ISPPROF and invoke ISPF
- If existing LOGON CLIST or EXEC customizes user's ISPF environment, remove that code and have user set up their own ISPF environment
- Educate users to Rexx EXEC or CLIST with ALTLIB and LIBDEF commands to customize their ISPF environment

Setting up TSO and ISPF (WAC)

- ISR@PRIM modified with "A" option for global applications, and "Z" option for applications specific to particular group

```
Menu Utilities Compilers Options Status Help
-----
                                ISPF Primary Option Menu
Option ===>

0 Settings      Terminal and user parameters      <   Calendar   >
1 View          Display source data or listings           July      2015
2 Edit          Create or change source data             Su Mo Tu We Th Fr Sa
3 Utilities     Perform utility functions                1  2  3  4
4 Foreground    Interactive language processing          5  6  7  8  9 10 11
5 Batch         Submit job for language processing        12 13 14 15 16 17 18
6 Command       Enter TSO or Workstation commands        19 20 21 22 23 24 25
7 Dialog Test   Perform dialog testing                   26 27 28 29 30 31
9 IBM Products  IBM program development products
10 SCLM         SW Configuration Library Manager        Time . . . . : 21:36
11 Workplace    ISPF Object/Action Workplace            Day of year. :   193
A ALL         ISPF Applications for all users
S SDSF         Spool Display and Search Facility
Z TSOAPPL     ISPF Applications for TSOAPPL group
```

Enter X to Terminate using log/list defaults

Setting up TSO and ISPF (WAC)

- The “Z” option is variable &ZSYSPROC

```
&ZQ = &Z
IF (&ZCMD ≠ ' ')
  &ZQ = TRUNC(&ZCMD, '.')
  &ZTRAIL = .TRAIL
  IF (&ZQ = ' ')
    .MSG = ISRU000
&ZSEL = TRANS ( &ZQ
0, 'PGM(ISPISM) SCRNAME(SETTINGS) '
1, 'PGM(ISRBRO) PARM(ISRBRO01) SCRNAME(VIEW) '
2, 'PGM(ISREDIT) PARM(P,ISREDM01) SCRNAME(EDIT) '
3, 'PANEL(ISRUTIL) SCRNAME(UTIL) '
4, 'PANEL(ISRFPA) SCRNAME(FOREGRND) '
5, 'PGM(ISRJB1) PARM(ISRJPA) SCRNAME(BATCH) NOCHECK '
6, 'PGM(ISRPTC) SCRNAME(CMD) '
7, 'PGM(ISPYXDR) PARM(&ZTAPPLID) SCRNAME(DTEST) NOCHECK '
8, 'PANEL(ISRLPRIM) SCRNAME(LMF) '
9, 'PANEL(ISRDIIS) ADDPOP '
10, 'PGM(ISRSCLM) SCRNAME(SCLM) NOCHECK '
11, 'PGM(ISRUDA) PARM(ISRWORK) SCRNAME(WORK) '
A, 'PANEL(ALLUSER) SCRNAME(ALLUSER) '
S, 'CMD(%@SDSF &ZTRAIL) NOCHECK '
Z, 'PANEL(&ZSYSPROC) SCRNAME(&ZSYSPROC) '
X,EXIT
SP, 'PGM(ISPSAM) PARM(PNS) '
' , ' '
*, '? ' )
```



Setting up TSO and ISPF (WAC)

- `&ZSYSPROC` holds member name of LOGON PROC, which enables defining a single proc with aliases
- This menu separates ISPF applications by function and presents single, standardized `ISR@PRIM` to all users



Setting up TSO and ISPF (WAC)

- After setting up LOGON PROC and Primary Option Menu, set up ALLUSER menu and each individual &ZSYSPROC menu for functional groups
- Dynamic ISPF Starter Set contains samples for LOGON PROC, LOGON EXEC, Primary Option Menu, ALLUSER menu, and &ZSYSPROC menu



Summary

- Miscellaneous recommendations made covering many topics important to Dynamic ISPF
- Selling Dynamic ISPF to Management can be done by pointing out performance, maintenance, disaster recovery, and security benefits of Dynamic ISPF



Summary

- ServerPac method merging all ISPF datasets into one large dataset prevents securing ISPF applications and favors “Standard” ISPF over Dynamic ISPF
- Roadmap for converting to Dynamic ISPF was discussed with test and production cutover plans



Summary

- The Dynamic ISPF Starter Set (DISS) is set of pre-coded Rexx driver EXECs designed to get you up and running quickly with Dynamic ISPF
- Setting up TSO and ISPF using WAC (World According to Conley) was discussed



Finally...

- IBM and third-party vendors have come around to Dynamic ISPF
- Dynamic ISPF support isn't an issue anymore
- If you encounter resistance, let me know and Dynamic ISPF community will respond with PMR's
- Properly coded ISPF application should **ALWAYS** support Dynamic ISPF



Finally...

- Read my articles in Cheryl Watson's TUNING Letter:
 - "Focus: Dynamic ISPF", 2000, No. 6
 - "Dynamic ISPF: An Update", 2003, No. 3
- Go to <http://www.watsonwalker.com> for details



Finally...

- Send me an Email at pincons@rochester.rr.com to get on the Dynamic ISPF mailing list
- If you implement Dynamic ISPF, please let me know your experiences, positive or negative; problems can be overcome, so let me know if you need assistance