

Logging on to TSO with ISPF on Multiple LPARs in the Same Sysplex with the Same Userid Under z/OS 1.4+

With the availability of z/OS 1.4, IBM has removed one of the restrictions that prevented using the same TSO userid on more than one system within the same JES2 MAS or Sysplex. It is now possible to logon to TSO on more than one LPAR in the same JES2 MAS or Sysplex with the same TSO userid. ISPF does not support this new capability, so while you can logon to TSO on multiple systems, you can only use ISPF on one of them. Lionel Dyck presents several options you have in dealing with these problems.

By Lionel B. Dyck

OVERVIEW

With the availability of z/OS 1.4 IBM has removed one of the restrictions that prevented using the same TSO userid on more than one system within the same JES2 MAS or Sysplex. The SYSIKJUA ENQ scope should be SYSTEM or RNL(INCL), not SYSTEMS or RNL(EXCL), and the JES2 code that restricted TSO logons to a single instance has been deleted (two lines of code).

With these changes it is now possible to logon to TSO on more than one LPAR in the same JES2 MAS or Sysplex with the same TSO userid.

The biggest **problem** with this is that ISPF does not support this new capability, so while you can logon to TSO on multiple systems, you can only use ISPF on one of them.

Another minor problem with multiple TSO logons is the user needs to be aware that notifications are sent to random logons and thus may appear on any of the logged-on sessions.

The first thing that must be done is to change ISPF so that the temporary data sets allocated (TEMP, CNTL, LIST, LOG) are unique to each LPAR. This can be easily accomplished by coding the ISPF Exit 16 to insert the four characters of the SMFID into one of the levels of the data set names (see FIGURE 1 for a sample SMP/E Usermod). With z/OS 1.5 ISPF there is a new configuration option that will eliminate the need to code Exit 16 for this purpose.

Note that IBM provides a sample ISPF Exit 16 in SYS1.SAMPLIB within member RACEXITS, which will insert the user's SECLABEL into the generated data set names.

The next step is more involved, as it concerns the ISPF Profile data set. This data set is a partitioned data set (PDS) in which ISPF saves profile information. A PDS can be shared for read access across multiple systems with no problems. Problems can arise if the PDS is shared across multiple systems in read/write/update mode. Since several systems can easily open the same member for update purposes and each can save that member, only the last save will 'win' with all previous saves being lost. Thus an update to a profile on system 'A,' which is saved after an update from system 'B,' will be the profile used on system 'B' the next time the user is on system 'B.' The result may be a surprise, as they will have a profile different from the one they saved. Thus the challenge for which there is no good solution.

One way to prevent this 'corruption' from occurring is to always allocate the ISPF Profile with DISP=OLD.

Since the ISPF Profile is allocated outside of ISPF, the installation has full control over the name and allocation of this data set.

There are several scenarios that can be considered, each with its own potential problems. The only perfect solution is for IBM ISPF Development to architect a shared ISPF Profile solution.

ACKNOWLEDGEMENTS

Bob Break of SBC Communications gave an excellent presentation at the SHARE meeting in Washington, D.C. in August 2003 during the JES2 Short Subjects session on how to share the ISPF Profile prior to z/OS 1.4, and many of his comments are still applicable.

Jim Narramore of Bank of America and SHARE ISPF Project Manager reviewed this document and provided several additional

FIGURE 1: SAMPLE SMP/E USER MODIFICATION FOR ISPF EXIT 16

```
//Jobname JOB . . . .
//          CLASS=D,MSGCLASS=X,NOTIFY=&SYSUID,TIME=1439
/*JOBPARM L=999999
/*****
//PU025AA EXEC SMPES
//SMPCSI DD DISP=SHR,DSN=SMPE.ZOS14.GLOBAL.CSI
//SMPCNTL DD *
          SET BDY(GLOBAL).
          REJECT SELECT(PU025AC) BYPASS(APPLYCHECK,ACCEPTCHECK).
          RESETRC.
          RECEIVE SELECT(PU025AC) SYSMODS LIST.
          SET BDY(S10DR40).
          APPLY SELECT(PU025AC) REDO ASSEM .
/*
//SMPPTFIN DD DATA,DLM='##'
++USERMOD (PU025AC) .
++VER(Z038) FMID(spf-fmid).
++SRC(ISPXDT) DISTLIB(AISPSAMP).
*****
*
* THIS MODULE SHOULD BE ASSEMBLED (OR THE EXITS POINTS
* ADDED TO AN EXISTING ISPEXITS MEMBER AND ASSEMBLED)
* WITH THE ISP.VXRXX.MACS DATASET AS THE MACLIB. THE
* RESULTING MODULE SHOULD BE LINKED (RENT AND REUS) AS
* MODULE ISPEXITS, WITH MODULE DYNAINSPF INCLUDED.
*
*
*          SPLEVEL SET=1                                R240
*          ISPMXED START
*
*          ISPMXDEF 16
*          ISPMPEPT ISPFEX16
*          ISPMXEND
*
*          ISPMXED END
*
*          ISPMXDD START
*          ISPMDDAD DYNAAAREA,256
*          ISPMXDD END
*          END
++SRC(ISPFEX16) DISTLIB(AISPSAMP).
*****
*
* MODULE NAME           : ISPFEX16
* ENTRY POINT NAME     : ISPFEX16
* ATTRIBUTES           : RENT,AMODE(31),RMODE(ANY)
* DESCRIPTIVE NAME     : ISPF EXIT 16 - TEMPORARY DATASET NAME EXIT
* FUNCTION              : MODIFY THE PREFIX OF ISPF-ALLOCATED TEMPORARY
*                        DATASETS BY ADDING THE SYSTEM-ID
*                        REF:ISPF/PDF PLANNING & CUSTOMIZING SC34-4257-01
*
* PARAMETERS           : R1 - PARAMATER LIST
*                        THIS IS DETAILED IN THE VOLUME ABOVE
*                        THERE IS A STANDARD LIST USED BY ALL ISPF EXITS
*                        FOLLOWED BY AN EXIT 16-SPECIFIC LIST
*
* RETURN CODES         : 0 (THIS IS IGNORED BY THE CALLER)
* CALLING MODULES      : ISPXDT
* CALLED MODULES       : NONE
* MACROS USED          : CVT      (MAPPING MACRO)
* MESSAGES ISSUED      : ?
*
*****
*
* REGISTER USAGE:
*
* R0 - ?
* R1 - PARAMETER LIST ADDRESS ON ENTRY
* R2 - ?
* R3 - ?
* R4 - ?
* R5 - ?
* R6 - ?
* R7 - ?
* R8 - ?
* R9 - POINTER TO CVT WHILE SEARCHING FOR SMF-ID
* R10 - PARAMETER LIST ADDRESS
* R11 - DSECT ADDRESS
```

Continued on next page.

points and improved the accuracy and completeness of this article.

Tom Conley, of Pinnacle Consulting and SHARE ISPF Deputy Project Manager, actively participated in discussions about this and provided several worthwhile comments and corrections to this article.

Thanks to all of those attending the SHARE meeting in Long Beach in February 2004 who participated in the discussion on this topic during the ISPF Sunday session.

SCENARIOS

Scenario 1: Permanent ISPF Profiles Unique to Each System

This is perhaps the easiest option, as the TSO Logon Proc can be set up to allocate the ISPPROF DD to a data set that is unique to the current system. Since the system symbolics are not supported in the TSO Logon Proc, this will mean a unique Proc for each system (either a unique name if in a shared Proc library or having multiple copies of the Proc customized in a Proc library for each system).

The issues with this scenario are:

1. The ISPF Profiles will be unique to each system. Changes made on one system will not be reflected on any other system.
2. Today the user expects the profile to be the same from logon to logon regardless of the system being used (assuming all systems share the same DASD and thus the same ISPF Profile). This will no longer be true.
3. The transition can be easy by copying the existing Profile data set into new system-specific data sets for the user.
4. Additional DASD space is now required and additional data sets to manage.
5. There is a higher probability that the user's ISPF Profile for any given system will be migrated when the user tries to use it (perhaps migrated to tape), which will delay the logon.

Scenario 2: Temporary ISPF Profile Data Set

For this scenario, the TSO Logon Proc will need to specify a PARM with a CLIST or REXX exec to be executed immediately after Logon and before any other processing occurs. In this code, the following steps should be coded:

1. Test using the SYSDSN function call to see if the standard ISPF data set exists

FIGURE 1: CONTINUED

```

*          R12 - BASE REGISTER
*          R13 - SAVE AREA POINTER
*          R14 - RETURN ADDRESS
*          R15 - ENTRY POINT ADDRESS (AT START)
*
*****
*
ISPFX16 CSECT
ISPFX16 AMODE 31
ISPFX16 RMODE ANY
        USING  ISPFX16,R15          ESTABLISH BASE REGISTER FOR CSECT
        B      ISPXSTRT             BRANCH ROUND ID
        DC     C'ISPFX16'           CSECT NAME
        DC     C'&SYSDATE'         ASSEMBLY DATE
        DC     C'&SYSTEMIME'       ASSEMBLY TIME
        DS     OF                   ALIGNMENT
ISPXSTRT EQU *
        STM    R14,R12,12(R13)     SAVE CALLERS REGS IN CALLERS SVAREA
        LR     R12,R15             SET UP BASE REGISTER
        DROP   R15
        USING  ISPFX16,R12         ESTABLISH BASE REGISTER
        LR     R10,R1              SAVE PARAMETER LIST ADDRESS
*
        GETMAIN RC,LV=WRKDSLEN,SP=0,LOC=BELOW
        LR     R11,R1              SAVE ADDRESS OF STORAGE
        USING  WRKDSCT,R11        ANCHOR DSECT
*
        LA     R14,SAVEAREA        PUT CALLER SVAREA ADDR IN OWN SVAREA
        ST     R14,8(R13)          SET FORWARD POINTER
        ST     R13,SAVEAREA+4      SET BACK CHAIN TO CALLER
        LA     R13,SAVEAREA        POINTER TO SAVE AREA
        EJECT
*
ISPXMAIN EQU *
*
        L      R9,CVTPTR           GET THE SYSTEM ID
        USING  CVT,R9
        L      R5,CVTSMCA
*
        L      R8,28(R10)          ADDRESS OF PREFIX
        L      R7,24(R10)          ADDRESS OF PREFIX LENGTH
        LR     R6,R8               ADDRESS OF PREFIX
        A      R6,0(0,R7)          PLUS PREFIX LENGTH
        MVI    0(R6),C'.'         ADD A DOT
        MVC    1(4,R6),16(R5)     AND THE SYSTEM ID
*
        L      R5,0(0,R7)          PUT PREFIX LENGTH IN R5
        A      R5,=F'5'           ADD 5 TO IT
        ST     R5,0(0,R7)         AND STORE IT BACK AGAIN
*
*****
*          RETURN ROUTINE - RELEASE AREAS, RESTORE REGS, RETURN TO CALLER
*
*****
*
ISPXEND EQU *
        L      R13,SAVEAREA+4     PICK UP BACK CHAIN POINTER
*
        FREEMAIN RC,LV=WRKDSLEN,A=(R11),SP=0
*
        L      R14,12(0,R13)       RESTORE R14
        LM     R0,R12,20(R13)      RESTORE R0-R12
        SR     R15,R15             SET ZERO RETURN CODE
        BR     R14                 RETURN TO CALLER
        EJECT
*
R0 EQU 0
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
R5 EQU 5
R6 EQU 6
R7 EQU 7

```

Continued on next page.

(allocate it new if it does not and then free it)

2. Allocate the standard ISPF Profile data set DISP=OLD
3. If the return code is 0 then re-allocate the data set with DISP=SHR and exit
 - a. The reallocation to DISP=SHR is to allow additional logons to access the standard ISPF Profile data set to make the temporary copy
4. If the return code is not 0 (12), then the data set is already allocated by some other user in the sysplex, so now:
 - a. Allocate a temporary ISPF Profile data set
 - b. Copy all members from the standard ISPF Profile data set into the newly allocated temporary data set
 - c. exit

At this point, a temporary ISPF Profile data set is allocated and available for ISPF to use.

The issues with this scenario are:

1. Any profile changes made using this temporary ISPF Profile will be lost when the user logs off.
2. Today the user expects the profile to be the same from logon to logon regardless of the system being used (assuming all systems share the same DASD and thus the same ISPF Profile). This will not be true any longer.

Scenario 3: Sharing a Single ISPF Profile Data Set

This is another option, one which some installations are using, but also one that has other challenges as well as benefits.

To do this scenario you need to:

1. Change the ENQ SPFEDIT from SCOPE=SYSTEMS to SCOPE=SYSTEM, but only for data sets which match your ISPF Profile data set (e.g. *.ISPF.PROFILE) using the GRS Wildcard facility.

That is simple enough, but there are issues with this technique:

1. If the profile data set goes into an additional extent while the user is logged on to system 'A' and system 'B,' the system(s) where the extra extent was not added will not be able to see any information in the new extent.

FIGURE 1: CONTINUED

```

R8      EQU      8
R9      EQU      9
R10     EQU      10
R11     EQU      11
R12     EQU      12
R13     EQU      13
R14     EQU      14
R15     EQU      15
*
          LTORG
*
WORKAREA DS      0D
*
WRKDSECT DSECT
*
SAVEAREA DS      18F
*
WRKDSLEN EQU      *-WRKDSECT
          CVT      DSECT=YES
          END
++JCLIN .
//LKED   EXEC      PGM=IEWL,
//        PARM='  NCAL,MAP,LIST,LET,RENT,REUS'
//SYSUT1 DD      UNIT=SYSDA,SPACE=(2048,(200,20))
//SYSPRINT DD     SYSOUT=*
//SISPLD DD      DISP=SHR,DSN=ISP.SISPLD
//SYSMOD DD      DISP=SHR,DSN=ISP.SISPLD
//SYSLIN DD      *
INCLUDE  SISPLD(ISPEXITS)
INCLUDE  AISPMOD1(ISPFEX16)
ENTRY   ISPXDT
NAME    ISPEXITS(R)
/*
###
  
```

Share Networks and Operating Environments

2. Assume the user is logged onto multiple systems; as the user logs off, or exits a specific application, the profiles are saved. This means that the last application exit or system off will 'win' with that save as it will overlay the profile member with its version of the profile.
3. It is possible that the user will make a profile change and then lose it when they logoff from another system after logging off the system where the profile change was made. This inconsistency is a change from what the users expect and what they are used to.

migrate beyond z/OS 1.3 you will have this challenge to deal with. Hopefully this article has provided you with the information that you will need to make the correct decision for your installation.



NaSPA member Lionel B. Dyck is a z/OS systems programmer for a large health maintenance organization in California. He has been in systems programming since 1972 and is an active participant at SHARE. lionel.b.dyck@kp.org is his e-mail address.

CONCLUSION

As you can see, you have several options to deal with the problems that IBM has created with their change to JES2 with z/OS 1.4. Sadly, with each option there are still issues that you and your users must be aware of until IBM steps up to the plate and provides a true ISPF Profile sharing capability. If none of these options are acceptable, then the best solution is to change the SYSIKJUA ENQ to a SYSTEMS, or RNL(EXCL), enqueue rather than a local one. In any case, once you