

An Experienced ISPF User Shares his Secrets

Bruce Koss
Wells Fargo

August 11, 2012
11559

Introduction

- Provide an overview on how to exploit several ISPF features
 - ISPF Configuration
 - Stacking commands
 - MODEL command
 - Work Station Agent (WSA)
- Expectations
 - Familiarity with ISPF Configuration and MODEL command
 - Capability to allocate personal ISPF datasets to allow testing
- Other Items
 - Links are provided to IBM manuals and other documents to assist in learning these features in more detail
 - May need to adjust Adobe Reader zoom value to view text

Agenda

- ISPF Configuration
 - Settings
- Stacking Commands
 - Overview
 - Implementation
- MODEL Command
 - Obtain ISPF commands
 - Add JCL Utilities, SDSF REXX and REXX commands
- Work Station Agent (WSA)
 - Overview and installation
 - Transfer files using ISPF 3.7.2

ISPF Configuration

Introduction

- Provide an overview of several ISPF configuration variables to
 - Decrease response time
 - For new users, eliminate time configuring environment
 - Provide new features/enhancements
- ISPF Configuration Dialog
 - Suggest executing this after installing a new z/OS operating system
 - Execution
 - ➔ [TSO ISPCCONF](#)
- Links
 - SHARE, Session 10948, March 14, 2012, Tom Conley
 - [Configuring ISPF for Fun and Profit](#)
 - Chapter 2, The ISPF Configuration Table
 - [ISPF – V1R13 – Planning and Customization](#)

ISPCCONF

Variable	Assigned	Notes
APPLID_FOR_SITE_COMMAND_TABLE	WFS	Allows for system/site wide command tables to be defined/used (WFSCMDS) - WFS stands for Wells Fargo Site commands - Default is NONE
APPLID_FOR_USER_COMMAND_TABLE	WFU,WFU2, WFU3	Allows users to define up to three different user command tables (WFUCMDS, WFU2CMDS and WFU3CMDS) - WFU stands for Wells Fargo User commands - Default is NONE
BLOCK_SIZE_FOR_TEMPORARY_CNTL_DATA_SETS BLOCK_SIZE_FOR_TEMPORARY_LIST_DATA_SETS BLOCK_SIZE_FOR_TEMPORARY_WORK_DATA_SETS	0 0 0	Set block size to zero to obtain/use half track blocking - Default is 800, 3146 and 2560
COMMAND_LINE_PLACEMENT	ASIS	Place command line on top - Default is BOTTOM
DEFAULT_LIBDEF_PROCESSING_OPTION	STACK	When LIBDEFing datasets, the default will be STACK - Prevents issues when stacking software on top one another - Default is UNCOND
DEFAULT_SYSTEM_NAME	ON	Displays lpar name in upper left hand corner of all panels - Default is OFF

ISPCCONF

Variable	Assigned	Notes
DISALLOW_WILDCARDS_IN_HLQ	YES	Prevents long enqueues on the master catalog - Default is NO
DSLIST_RM_ENABLED DSLIST_RM_COMMAND	YES %CA1DS34	Allows executing a REXX program to obtain CA1 information when typing an “I”, “S” or “M” next to a tape dataset from ISPF 3.4 - CA1DS34 is an in-house written utility -- “I” invokes CA1, selects option “1 - INQ/UPD” and inserts VOLSER to obtain tape information -- “S” invokes CA1's Browse Module (TSOTIQ) and uses dataset name to obtain tape information -- “M” invokes CA1's Tape Inquiry Program (TMSBINQ) and uses VOLSER to obtain tape info. - Can use CA1 REXX program TSOTIQ - Default is NO and %EDGRPD34
DISPLAY_EMPTY_MEMBER_LIST DISPLAY_EMPTY_MEMBER_LIST_PATTERN	YES YES	Capability to edit an empty PDS - Default is NO and NO
DISPLAY_SEQUENCE_NUMBERS	OFF	- Default is ON
EDIT_RECOVERY_BLOCK_SIZE EDIT_RECOVERY_PRIMARY_QUANTITY EDIT_RECOVERY_SECONDARY_QUANTITY	27998 800 800	Allow for large amount of changes and updates - Default is 13680 , 40 and 200

ISPCCONF

Variable	Assigned	Notes
HILITE_DOLOGIC HILITE_IFLOGIC HILITE_PAREN	ON ON ON	Distinguishes lines of code and assists in troubleshooting - Default is OFF, OFF and OFF
ISPCTL0_BLOCK_SIZE	0	Set block size to zero to obtain/use half track blocking - Default is 800
ISPF_TEMPORARY_DATA_SET_QUALIFIER	&SYSNAME.	Inserts the lpar name in the second level qualifier of the ISPF temporary dataset names - Allows logging on to multiple, concurrent TSOs within a sysplex - If lpar name begins with a number, suggest placing the character 'S' in front of &SYSNAME - Default is NONE
LIST_DATA_SET_RECORDS_PER_BLOCK	231	Set list data set records per block to 231 for a 3390 - LRECL 121 * 231 = 27951, half-track for a 3390 - Default is 26
LIST_SYSOUT_CLASS	*	- Default is NONE
LOCAL_LIST_SYSOUT_CLASS	*	- Default is NONE
LOG_DATA_SET_BLOCK_SIZE	0	Set block size to zero to obtain/use half track blocking - Default is 129

ISPCCONF

Variable	Assigned	Notes
LOG_DATA_SET_DISPOSITION	D	Delete log data set - Default is NONE
LOG/LIST_JOB_CARD1	Job Card	Defined a standard JOB card format - Software can obtain and use it by coding “VGET (ZLLGJOB1 ZLLGJOB2 ZLLGJOB3 ZLLGJOB4)” - Default is NONE
LOG_PAGES_PRIMARY_QUANTITY LOG_PAGES_SECONDARY_QUANTITY	0 0	Not needed/used, decreases response time - Default is 10 and 10
MAXIMUM_NUMBER_OF_SPLIT_SCREEN	32	Capability to have 32 split screens - Set PF2= START , PF9= SWAP NEXT and PF21= SWAP PREV , then activate SWAPBAR - Default is 8
NUMBER_OF_PFKEYS	12	Typing in KEYS will display “Number of Keys = 24 ” - Default is 12
OUTLIST_BLOCK_SIZE	0	Set Outlist Utility block size to zero to obtain half track blocking for OUTLIST datasets - Default is 13566
PROFILE_SHARING	YES	Allows logging on to multiple, concurrent TSOs within a sysplex - Required to run multiple, concurrent z/OSMF ISPF sessions - Default is NO

ISPCCONF



Variable	Assigned	Notes
SCROLL_DEFAULT	CSR	Set scroll default to cursor - Default is PAGE
SHOW_PFKEYS	OFF	Provides for more screen real-estate - Default is ON
SITE_WIDE_INITIAL_MACRO	RESET	Prevents unnecessary messages from appearing when editing a dataset or member - Default is NONE
STANDARD_NUMBERS	OFF	- Default is ON
STATS_EXT_ENABLED	YES	Extended statistics are kept for PDS members - Default is NO
TAB_TO_ACTION_BARS	NO	Pressing the HOME key will position cursor on primary command line - Default is YES
UNDO_STORAGE_SIZE	1024	Allocate 1024K of memory to edit UNDO command to be used for keeping history of edit changes - Enables SETUNDO command - Default is 0

ISPCCONF

Variable	Assigned	Notes
USE_KEYLISTS	NO	Makes PF keys standard through out ISPF - Assigns PF4=RETURN, PF5=RFIND, PF6=RCHANGE, P7=UP, PF8=DOWN, PF9=SWAP, PF10=LEFT, PF11=RIGHT and PF12=RETRIEVE - Default is YES
VSAM_BROWSE_ENABLED VSAM_BROWSE_COMMAND VSAM_EDIT_ENABLED VSAM_EDIT_COMMAND VSAM_VIEW_ENABLED VSAM_VIEW_COMMAND	YES %ISPF34B YES %ISPF34E YES %ISPF34B	Allows invoking a REXX program when browsing, editing or viewing a VSAM file - ISPF34B/E is a in-house written REXX program to invoke File-Aid/MVS (F1/2) or CBT183 (BR) - Can use File-Aid/MVS REXX programs F1 (Browse) and F2 (Edit) - Can install and invoke Ditto, File Manger or Review from CBT tape - Default is NO and FMNMAIN DSB /
WSA_DOWNLOAD_DATA_SET	'SYS1.SISPGUI'	Dataset name inserted into ISPF 3.7.1 screen - Default is 'ISP.SISPGUI'

Stacking Commands

Introduction



- Stacking commands provides the capability to invoke any application from anywhere within ISPF and not have to be located at the user or main menu
- Allows multiple software products to be stacked on one split screen and each split screen is a separate stack
- Stacking commands are implemented using the ISPF site and user command tables (slide 6)

Syntax

- All menu options are stacking commands
- IBM Primary Menu
 - Prefix the character “Z” in front of option (Z##)
 - Examples
 - Z1 will invoke ISPF option 1, Browse
 - Z2 will invoke ISPF option 2, Edit
 - Z34 will invoke ISPF option 3.4, Data Set List
 - ZSD will invoke ISPF option SD, SDSF
- Developer and System Menu
 - Type in the entry name
 - CA7, ABENDAID, NDM, OPSMVS, etc.
 - Only need to type in the 1st three or four characters
 - ABE for ABENDAID
 - OPS for OPSMVS

Fast Path

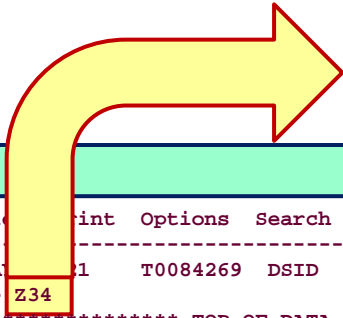
- To fast path directly into a particular menu entry
 - Use the ISPF delimiter character
 - Default is the semi-colon (;)
- Examples of fast pathing into a software product
 - SDSF, Status of jobs and set owner to R2221
 - `ZSD;ST;OWNER R2221` -or- → `ZSD ST;OWNER R2221`
 - OPSMVS, State Manager, Resource Status , option 4.11.2
 - `OPS;4;11;2` -or- → `OPS 4;11;2` -or- → `OPSMVS 4;11;2`
 - Endeavor option 2.1
 - `ENDE;2;1` -or- → `ENDE 2;1` -or- → `ENDEVOR 2;1`
 - SPUFI for DB2 subsystem DI32, option 1.1
 - `DB2U DI32;1;1`

How does it work?

- A user who is in SDSF may have a need to invoke ISPF 3.4, Data Set List Utility, for a quick look at a data set
- By simply keying in **Z34** on the command line, they can stack ISPF 3.4 on top, without interrupting the SDSF session or starting up a new logical split screen
- The **END** or **RETURN** PFkeys (PF3/4) will cause an immediate return to the suspended dialog, in this case SDSF

Example – Z34

```
SDSF
-----
Display Filter View Print Options Search P
-----
SDSF OUTPUT DISPLAY 1 T0084269 DSID
COMMAND INPUT ==> Z34
***** TOP OF DATA *****
1 //R2221 JOB 'TSOPRD',REGION=8192K
2 //PANTSOPG EXEC PANTSOPG
3 XXPANTSOPG EXEC PGM=IKJEFT01,
  XX DYNAMNBR=200,
  XX REGION=3072K,
  XX TIME=480,
  XX PARM='%XAPAN'
4 XXSTEPLIB DD DISP=SHR,DSN=PRDLL.#OU.VPS.LL2.LINKLIB
  XX*****
  XX*
  XX* ***** TSO PROCEDURE USED BY - S & P - *****
  XX*
  XX*****
  XX*ISPCTL1 DD SPACE=(CYL,(1,1)),UNIT=TTEMP,
  XX* DCB=(BLKSIZE=800,LRECL=80,RECFM=FB)
  XX*ISPCTL2 DD SPACE=(CYL,(1,1)),UNIT=TTEMP,
  XX* DCB=(BLKSIZE=800,LRECL=80,RECFM=FB)
5 XXISPLST1 DD SPACE=(CYL,(1,1)),UNIT=TTEMP,
  XX DCB=(BLKSIZE=1210,LRECL=121,RECFM=FBA)
```



```
Z34
-----
Menu RefList RefMode Utilities Help
-----
Data Set List Utility

Option ==>

blank Display data set list          P Print data set list
V Display VTOC information            PV Print VTOC information

Enter one or both of the parameters below:
Dsname Level . . . PRSS.R2221.JCL
Volume serial . .

Data set list options
Initial View
1 1. Volume
2 2. Space
3 3. Attrib
4 4. Total

Enter "/" to select option
/ Confirm Data Set Delete
/ Confirm Member Delete
/ Include Additional Qualifiers
/ Display Catalog Name
/ Display Total Tracks
/ Prefix Dsname Level

When the data set list is displayed, enter either:
"/" on the data set list command field for the command prompt pop-up,
an ISPF line command, the name of a TSO command, CLIST, or REXX exec, or
```

STACK – Utility

- An in-house utility called **STACK** or **STK** was developed to display all available stacking commands
- Users can type in **STACK** or **STK** on any primary command line
- Displays all available commands defined in site (**WFSCMDS**) or user (**WFU[2|3]CMDS**) commands table
- Lists all entries on the Developers, System and ISPF menus, plus TSO commands like **PING**, **NETSTAT**, etc.
- An easy way to locate, execute and obtain product information

STACK – Example

- Typing in **STACK** or **STK** displays all stacking commands

```
----- ( Stacking Commands )----- Row 1 to 11 of 320
Command ==>                               Scroll ==> CSR

Command:  F xxx - Find text      L xxx - Locate text      xxx  - Exec command
          O xxx - Only text      R      - Reset display   S xxx - Sort column

Line:     S - Invoke utility    B - Browse overview

Trunc:    Number of characters needed to invoke Stacking Command.
          If field is zero, must type in full command.

S Command  Trunc  Description
-----
- AAVIEW   3      AbendAid Hotkey
- ABENDAID 3      Dump Analysis Tool
- ACR34    0      Automated Controls and Reconciliation
- ACR35    0      Automated Controls and Reconciliation
- AMBLIST  3      Load Module Map and Cross Reference Listing
- APFLIST  4      APF Datasets
- APPC     0      Advanced Program To Program Communication
- ASID     0      Address Space Identifier (ASID) Usage Summary
- AUTOCMDS 5      OPSMVS - MVS Automation Cmds
- AXCIS    0      Administration
- BDT      0      Bulk Data Transfer
```

Pros & Cons

- Pros
 - Enhances productivity in ISPF
 - Allows invoking multiple applications on one screen
 - And on top of one another
 - "Stacking" capability is limited only by amount of TSO virtual storage
 - Reduces amount of time needed to invoke the facility
- Cons
 - When two or more applications are stacked on the same split screen
 - Can only access/use the active one
 - The one on the top of the stack
 - Can not select an application in the middle of the stack
 - Typing in “=??” will remove the entire stack
 - Where ?? is an ISPF menu entry
 - Examples: =3.4, =D.ABENDAID, =X, etc.

Implementation – ISPCCONF

- ISPF Configuration Utility
 - Type in
 - ➔ TSO ISPCCONF
 - Define/set
 - APPLID_FOR_SITE_COMMAND_TABLE = WFS
 - APPLID_FOR_USER_COMMAND_TABLE = WFU, WFU2, WFU3
- The user command table allows programmers to create their own personal or team stacking commands
 - WFUCMDS, WFU2CMDS and/or WFU3CMDS
- Ensure ISPCFIGU module, created by ISPCCONF, is located in
 - LPA or LNK List
 - or ISPLLIB (STEPLIB) allocated to TSO logon proc

Implementation – ISPF 3.9

- Next few slides will demonstrate how to create a site command table (WFSCMDS) using ISPF 3.9, Commands
- This same process can be used to create user commands (WFUCMDS, WFU2CMDS and WFU3CMDS)
- Prerequisite
 - Must allocate a PDS dataset to ISPTABL DD card
 - This is where the site command table will be saved (WFSCMDS)
 - Preferably a dataset concatenated to ISPTLIB DD card
 - Allows easy testing and validating of site commands
 - Dataset attributes: RECFM=FB,LRECL=80

Implementation – ISPF 3.9

- Invoke ISPF 3.9, insert site command table name ([WFS](#))

```
Command Table Utility

Command ==>

Specifications
Application ID . . WFS
Enter "/" to select option
    Show description field

Command table search order
Application table . : ISP
User table 1 . . . . :
User table 2 . . . . :
User table 3 . . . . :
Site table 1 . . . . :
Site table 2 . . . . :
Site table 3 . . . . :
System table . . . . : ISP

If no application ID is specified, the current application ID will be
used. The name of the command table to be processed is formed by
prefixing the application id to the string 'CMDS'. For example:
Application ID . . TST results in a command table name of 'TSTCMDS'.
```

Implementation – ISPF 3.9

- Tab down to an entry and type in “E” for edit

```
File  Menu  Utilities  Help
-----
                                Update WFSCMDS                Row 1 to 15 of 17
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
  _____  -
E   _____  -
  _____  -
  _____  -
  _____  -
  _____  -
  _____  -
  _____  -
  _____  -
  _____  -
  _____  -
  _____  -
  _____  -
  _____  -
  _____  -
  _____  -
  _____  -
```


Implementation – ISPF 3.9

- Insert Verb (stacking command), Trunc (number of chars needed), Action (what to execute/invoke) and Description

```
File  Menu  Utilities  Help
----- Command Table Utility -----
                          Extended Command Entry

Command ==>

Make changes to the command and select Update to update the entry or
Cancel to ignore the changes.

Verb   . . .  Z34
Trunc  . . .  0
Action . . .  SELECT PGM(ISRUDL) PARM(ISRUDLP) NEWAPPL(ISR)

Description  ISPF 3.4 - Utilities - DSLIST

Enter / to select option
_ Allow mixed-case in Action field

                          Update                          Cancel
```

Implementation – ISPF 3.9

- VERB
 - Stacking command
 - Must follow IBM naming convention
 - 1st char = A-Z \$#@ and 2nd-8th char = A-Z 0-9 \$#@
 - Must be 2 to 8 characters in length
- TRUNC
 - Minimum number of characters the user must enter
 - If zero or equal to verb length, user must enter entire command
- ACTION
 - SELECT command, program, or selection panel to be given control
- Description
 - Brief description/purpose of command
 - 80 characters or less

Implementation – ISPF 3.9

- Repeat the process for all menu entries
- Terminate/re-invoke ISPF and test typing in a stacking command
- Copy the site command table (**WFSCMDS**) into an ISPTLIB dataset accessible to everyone
- To update the active site command table (**WFSCMDS**)
 - Need to make a copy (ex: **WFSNCMDS**)
 - Use ISPF 3.9 to make changes (**WFSNCMDS**)
 - Copy with replace (**WFSNCMDS** → **WFSCMDS**)
 - Terminate/re-invoke ISPF and validate

Suggestions

- Always define site commands with three or more characters
 - Stacking commands take precedence over menu options
 - Prevents menu conflicts
- Prefix the character “Z” in front of all IBM Primary Menu options
 - Very few selection menus use character Z and a number (Z1, Z2, etc.)
- Ensure software LIBDEFing datasets use sub parameter **STACK**
 - Change default LIBDEF processing option to **STACK**
 - Do not use LIBDEF sub parameter **UNCOND**

Suggestions

- To prevent S878 abends, due to several split screens and stacks
 - Increase TSO logon size to 8 megabytes (8192K) or above the line
 - Set REXX/CLIST variable storage above the line
 - ➔ `TSO PROFILE VARSTORAGE(HIGH)`
- Use ISPF 3.9, Create/change application command table
 - Specifically designed for updating site and user command tables
 - Where ISPF 3.16, ISPF table utility, is a general purpose utility
 - Used to update a wide variety/range of tables

Notes

- Command table search order

Order	Command Table	Member
1	User command table 1	WFUCMDS
2	User command table 2	WFU2CMDS
3	User command table 3	WFU3CMDS
4	Site command table 1	WFSCMDS
5	Site command table 2	
6	Site command table 3	
7	System table	ISPCMDS

IVPs

- Some Independent Vendor Products (IVPs) provide the ability to implement stacking commands
- CA – PDSMAN
 - Initialization Parameter
 - Use \$EZYCMD to define site commands
 - Pros
 - Can refresh site command table by executing below MVS command
→ F PDSMAN,REFRESH
 - Cons
 - To test/validate, must use a different lpar from where the general TSO population is logged onto
 - Must terminate/re-invoke ISPF to access new commands
 - Does not allow user command tables

CBT Tape

- Several command table utilities can be found on CBT Tape
- Link
 - [CBT Downloads](#)
- Files
 - [183 – Gilbert Saint-flour’s collection](#)
 - FASTPATH – Another way to create/use a site command table
 - [312 – Lionel Dyck Collection of Utilities. A thru R](#)
 - ISPFMDS – Dynamically update ISPF Command Table
 - [349 – REXX to update in core ISPF command table on the fly](#)
 - Prevents having to exit and re-invoke ISPF
 - [578 – General ISPF Table Handling Facility – Roy Gardiner](#)

Docs

- Technical Support, 2003, Lionel Dyck
 - [ISPF Command Tables for the Average Users](#)
- Using command tables to define commands
 - [ISPF – V1R13 – Users Guide – Vol 1](#)

MODEL Command

Introduction

- MODEL command
 - IBM edit macro
 - Quickly/easily obtain skeleton ISPF commands
 - Most useful, but least known/utilized
 - Continually being updated/enhanced by IBM
 - Can only be used in EDIT and VIEW mode
- Links
 - SHARE, Session 11567, Aug 2012, Liam Doherty and Peter Van Dyke
 - [ISPF Editor – Beyond the Basics, Hands-On Lab](#)
 - SHARE, Session 2646, August 22, 2002, Lionel B. Dyck
 - [ISPF Design Coding Hints and Tips](#)
 - Chapter 4, Using edit models
 - [ISPF – V1R13 – Edit and Edit Macros](#)

Enhanced

- MODEL command can be enhanced to provide
 - IBM, OEM and in-house JCL skeletons
 - SDSF REXX examples
 - REXX commands
- Next few slides display
 - Changes highlighted in the IBM ISREMCLS panel
 - ISRREMJCL panel containing table of contents of all JCL utilities
 - Sample JCL skeletons, MIEHLIST and MUTKT
 - Changes highlighted in the IBM ISREMRXC panel
 - Where to obtain SDSF REXX and REXX commands

JCL – ISREMCLS – Panel

Model Classes

Option ==>

Enter number or Class of model.

Enter END command to cancel MODEL command.

- 1 CLIST - ISPF services in CLIST commands
- 2 COBOL - ISPF services in COBOL programs
- 3 EXEC - ISPF services in EXEC commands
- 4 FORTRAN - ISPF services in FORTRAN programs
- 5 MSGS - Message format
- 6 PANELS - Panel formats and statements
- 7 PLI - ISPF services in PLI programs
- 8 SKELS - File tailoring control statements
- 9 PASCAL - ISPF services in PASCAL programs
- 10 REXX - SDSF function and ISPF services in REXX execs
- 11 DTL - ISPF Dialog Tag Language formats and statements
- 12 C - ISPF services in C/370 programs
- 13 SCLM - SCLM Project Definition Macros
- 14 ARCHDEF - SCLM Architecture Definition templates
- 15 JCL - JCL for selected utility programs (USER-ADDED)

JCL – ISREMCLS – Panel

```

)PANEL KEYLIST(ISRSNAB,ISR)
/*-----*/
/* FLG  YMMDD  USERID  DESCRIPTION  */
/*-----*/
/* @A2  100519  R2221   Last level qual. = JCL, CNTL, PROC, INST */
/* @A1  100203  R2221   Added .ISPCLIST, .ISPSEX, and .ISPEXEC */
/* @A0  090118  R2221   Added Option Number 15 - JCL utilities */
/* @A0  090118  R2221   Find everything that was modified execute: */
/* @A0  090118  R2221   ==> X ALL;F '15' ALL;F 'ISREMJCL' ALL */
/*-----*/
)ATTR DEFAULT( ) FORMAT(MIX) /* ISREMCLS - ENGLISH - 5.8 */
0D TYPE(PS)
05 TYPE(PT)
06 TYPE(PIN)
09 TYPE(FP)
0A TYPE(NT)
11 TYPE(SAC)
10 TYPE(ET)
27 TYPE(NEF) CAPS(ON) PADC(USER)
28 AREA(SCRL) EXTEND(ON)
)BODY WINDOW(76,22) EXPAND(..) CMD(ZCMD)
.. .Model Classes. ..
.Option ==>.Z. .
.
.Enter number or Class of model.
.Enter.END.command to cancel MODEL command.
.SAREA40
)AREA SAREA40
.1 .CLIST  - ISPF services in CLIST commands
.2 .COBOL  - ISPF services in COBOL programs
.3 .EXEC   - ISPF services in EXEC commands
.4 .FORTRAN - ISPF services in FORTRAN programs
.5 .MSGS   - Message format
.6 .PANELS - Panel formats and statements
.7 .PLI    - ISPF services in PLI programs
.8 .SKELS  - File tailoring control statements
.9 .PASCAL - ISPF services in PASCAL programs
.10.REXX   - SDSF function and ISPF services in REXX execs
.11.DTL    - ISPF Dialog Tag Language formats and statements
.12.C     - ISPF services in C/370 programs
.13.SCLM  - SCLM Project Definition Macros
.14.ARCHDEF - SCLM Architecture Definition templates
.15.JCL   - JCL for selected utility programs (USER-ADDED)
)INIT
.ZVARS = 'ZCMD'
&ZWINTTL = ' '
.HELP = ISR2P100
.MSG = ' ' /* BLANK MSG INITIALLY */
&ISRMCLAS = ' ' /* AND THE CLASS HOLDER VARIABLE */
.CURSOR = 'ZCMD'

```

```

)PROC
IF (&ISRMONCL = 'Y') /* ARE WE ONLY OBTAINING A CLASS ? */
&ISRMCLAS = TRANS(TRUNC(&ZCMD, '.')) 1,CLIST 2,COB 3,EXEC 4,FORT
5,MSGS 6,PANELS 7,PLI 8,SKELS 9,PASCAL 10,REXX 11,DTL 12,C
13,SCLM 14,ARCHDEF *,*)
/* @H1C@H5C*/
&TMPTRAIL = .TRAIL /* Save any remaining @H1A*/
IF (&ISRMONCL = 'Y') /* ARE WE ONLY OBTAINING A CLASS ? OZ83844*/
&ISRMDSPL = ' '
&ZCMD = TRUNC (&ZCMD, '.')
&ZTMPTRL = .TRAIL
IF (&ZCMD = 'COBOL') &ZCMD = TRUNC(&ZCMD,3)
IF (&ZCMD = 'EXEC') &ZCMD = TRUNC(&ZCMD,4)
*REXX(ZCMD, JCLQUAL, CNTLQUAL, PROCQUAL, INSTQUAL) /* @A2 */
IF (POS("JCL",ZCMD) > 0) THEN /* @A2 */
JCLQUAL = ZCMD /* @A2 */
ELSE /* @A2 */
JCLQUAL = "JCL" /* @A2 */
IF (POS("CNTL",ZCMD) > 0) THEN /* @A2 */
CNTLQUAL = ZCMD /* @A2 */
ELSE /* @A2 */
CNTLQUAL = "CNTL" /* @A2 */
IF (POS("PROC",ZCMD) > 0) THEN /* @A2 */
PROCQUAL = ZCMD /* @A2 */
ELSE /* @A2 */
PROCQUAL = "PROC" /* @A2 */
IF (POS("INST",ZCMD) > 0) THEN /* @A2 */
INSTQUAL = ZCMD /* @A2 */
ELSE /* @A2 */
INSTQUAL = "INST" /* @A2 */
*ENDREXX
&ZCMD = TRANS (&ZCMD
CLIST,1 COB,2 EXEC,3 FORTRAN,4 FTN,4A FORT,4B MSGS,5 ISPMLIB,5A PANELS,6
MENUS,6A ISPLIB,6B PLI,7 PLI,7A PLIOPT,7B PLS,7C SKELS,8 PROCS,8A ISPLIB,8B
PASCAL,9 REXX,10 DTL,11 C,12 CPP,1C SCLM,13 ARCHDEF,14 JCL,15 ' ',' ','*')
IF (&ZCMD = ' ') &ZCMD = '&ZCMD.&ZTMPTRL'
&ZSEL = TRANS (TRUNC (&ZCMD, '.'))
1, 'PANEL(ISRECMCF)'
2, 'PANEL(ISREMCMB)'
3, 'PANEL(ISREMECX)'
4, 'PANEL(ISREMFN)'
4A, 'PANEL(ISREMFN)'
4B, 'PANEL(ISREMFN)'
5, 'PGM(ISRECMBR) PARM(ISREMMSG NOSEQ)'
5A, 'PGM(ISRECMBR) PARM(ISREMMSG NOSEQ)'
6, 'PANEL(ISREMPNL)'
6A, 'PANEL(ISREMPNL)'
6B, 'PANEL(ISREMPNL)'
7, 'PANEL(ISREMPLI)'
7A, 'PANEL(ISREMPLI)'

```

JCL – ISREMCLS – Panel

```
7B,'PANEL(ISREMPLI)'
7C,'PANEL(ISREMPLI)'
8,'PANEL(ISREMSKL)'
8A,'PANEL(ISREMSKL)'
8B,'PANEL(ISREMSKL)'
9,'PANEL(ISREMPAS)'
10,'PANEL(ISREMRXC)'
11,'PANEL(ISREMDTL)'
12,'PANEL(ISREMWLN)'
1C,'PANEL(ISREMWLN)'
13,'PANEL(ISREMFLM)'
14,'PANEL(ISREMARC)'
15,'PANEL(ISREMJCL)'
&JCLQUAL,'PANEL(ISREMJCL)' /* @A2 */
&CNTLQUAL,'PANEL(ISREMJCL)' /* @A2 */
&PROCQUAL,'PANEL(ISREMJCL)' /* @A2 */
&INSTQUAL,'PANEL(ISREMJCL)' /* @A0 */
ISPLIST,'PANEL(ISREMCMF)' /* @A1 */
ISPREXX,'PANEL(ISREMRXC)' /* @A1 */
ISPEXEC,'PANEL(ISREMRXC)' /* @A1 */
',',',',
*,'?'
IF (&ZSEL = '?')
  IF (&ISRMDSPL = 'RETURN') ** REPLACED BY OY33372*/
    IF (&ZMACRO = 'N') /* OY33372*/
      &ZEMTEMP = &ZCMD /* SET VARIABLE FOR MSG OZ91267*/
      .MSG = ISRYM012
      &ZCMD = ' ' /* RESET CMD LINE = BLANK OZ91267*/
      .RESP = 'ENTER' /* WANT TO RESET RESP FOR RE-ENT */
      /* IF (&ISRMDSPL = 'RETURN') ** REPLACED BY OY33372*/
    ELSE
      .RESP = 'END'
      &ISRMCLAS = '?'
      .MSG = ISRYM012 /* INVALID CLASS OY33372*/
&ISRMEND = 'N' /* SET THE END INDICATOR TO NO */
IF (.RESP = END) /* DID WE GET AN END FROM WHERE */
  IF (&ISRMDSPL = 'RETURN ') /* WAS IT BECAUSE WE HAVE MEMBER */
    &ISRMEND = 'Y' /* ITS BECAUSE USER HIT END KEY */
IF (&ISRMONCL = 'Y') /* ARE WE ONLY OBTAINING A CLASS */
  IF (.RESP = 'RETURN ') /* Has user hit extended return */
IF (&ISRMCLAS = ' ') /* Make sure we have some class */
  IF (&TMPTRAIL = ' ') /* Just class entered @H1A*/
    .RESP = END /* ONLY WANT CLASS, NO OTHER DISP.*/
  IF (&TMPTRAIL = ' ') /* More than class entered @H1A*/
    &ZEMTEMP = &ZCMD /* Set variable for message @H1A*/
    .MSG = ISRYM012 /* Set message number @H1A*/
    &ZCMD = ' ' /* Reset cmd lint = blank @H1A*/
    .RESP = 'ENTER' /* Reset RESP to Re-enter @H1A*/
&ZMSG = .MSG /* SET MESSAGE VARIABLE @L1A*/
VPUT(ZMSG) PROFILE /* SAVE MESSAGE IN PROF POOL @L1A*/
```

```
)PNTS
FIELD(ZPS01001) VAR(ZCMD) VAL(1)
FIELD(ZPS01002) VAR(ZCMD) VAL(2)
FIELD(ZPS01003) VAR(ZCMD) VAL(3)
FIELD(ZPS01004) VAR(ZCMD) VAL(4)
FIELD(ZPS01005) VAR(ZCMD) VAL(5)
FIELD(ZPS01006) VAR(ZCMD) VAL(6)
FIELD(ZPS01007) VAR(ZCMD) VAL(7)
FIELD(ZPS01008) VAR(ZCMD) VAL(8)
FIELD(ZPS01009) VAR(ZCMD) VAL(9)
FIELD(ZPS01010) VAR(ZCMD) VAL(10)
FIELD(ZPS01011) VAR(ZCMD) VAL(11)
FIELD(ZPS01012) VAR(ZCMD) VAL(12)
FIELD(ZPS01013) VAR(ZCMD) VAL(13)
FIELD(ZPS01014) VAR(ZCMD) VAL(14)
FIELD(ZPS01015) VAR(ZCMD) VAL(15) /* @A0 */
)END
/* 5694-A01 (C) COPYRIGHT IBM CORP 1980, 2005 */
/* ISPD TLC Release: 5.8. Level: PID */
/* z/OS 01.08.00. Created - Date: 24 Oct 2005, Time: 13:16 */
```

JCL – ISREMJCL – Panel

JCL Utility Models

Option ==>

Enter an option or move the cursor on JCL utility and press ENTER.
Enter END command to cancel MODEL command.

More: +

IBM		OEM	
I1	AMATERSE - Terse a file	O1	AFCLOGR1 - Extract OPERLOG records
I2	AMBLIST - Print linkage attributes	O2	CAL2X2WB - Execute CA7 Commands
I3	BPXBATC - Run USS shell scripts	O3	COB2EZT - Conv. COBOL to Eztplus
I4	CPPUPDTE - Search/update	O4	COMPAREB - Compare two files
I5	DFDSS - Backup/restore files	O5	CONSOLE - Execute MVS commands
I6	DFSORT - Sort/merge files	O6	COPYCAT - Copy a tape
I7	ICETOOL - Sort/merge files	O7	DAF - Dataset Audit Facility
I8	ICKDSF - Init disk pack	O8	DJCTEST - Abend job or set RC
I9	IDCAMS - Def/del/alter VSAM/GDGs	O9	EXAMBATX - CA Examine Compare
I10	IEBCOPY - Copy/del/move PDS mbr	O10	FCSWAIT - Wait a duration of time
I11	IEBGENER - Copy sequential files	O11	FILEAID - Copy/update/print files
I12	IEBPTPCH - Print and punch	O12	FTP - Transmit a file
I13	IEBUPDTE - Create/alter PDS mbr	O13	FTPSEC - Transmit a file secured
I14	IEHLIST - List PDS directory	O14	GENMAIL - Email using IEBGENER
I15	IEHPRGM - Rename/del/cat file	O15	LISTF - List DASD volumes
I16	IEFBR14 - Create/delete file	O16	MZCCMPB1 - Event Action Compare

JCL – ISREMJCL – Panel

JCL Utility Models

Option ==>

Enter an option or move the cursor on JCL utility and press ENTER.
Enter END command to cancel MODEL command.

More: - +

I16 IEFBR14	- Create/delete file	O16 MZCCMPB1	- Event Action Compare
I17 IFASMFDP	- Dump SMF records	O17 PDS	- PDS Utility
I18 IFCEREP1	- Dump LOGREC	O18 PDSEASY	- Batch Ezyedit
I19 IKJEFT01	- TSO batch	O19 PDSM13	- Library map/analysis
I20 IMASPZAP	- Zap/dump a module	O20 PDSM18	- String scan/replace
I21 IRXJCL	- Execute REXX program	O21 PDSM25	- Member size analysis
I22 ISRSUPC	- Search/compare	O22 PDSM32	- Delete generation mbrs
I23 REXXCL	- Compile/link REXX pgm	O23 PDSM33	- Compare/update files
I24 TRSMAN	- Terse a file	O24 PDSM36	- Compare files
I25 SDSF	- SDSF batch	O25 PDSMERGE	- Merge PDS datasets
I26 SMP	- IBM's ChangeMan	O26 PKZIP	- Zip/unzip a file
JCL		O27 QWIKREF1	- MVS/Quick Reference
J1 COMMAND	- Command card	O28 SASSBSTR	- Execute CA7 commands
J2 DD	- DD card	O29 SAVRS	- Obtain syslog/reports
J3 JCLLIB	- Proclib card	O30 SDEBYSD	- Side by side comparison
J4 JOB	- Job card	O31 SHOWMVS	- IPL information
		O32 SYSWAIT	- Wait a duration of time

JCL – ISREM JCL – Panel

JCL Utility Models

Option ==>

Enter an option or move the cursor on JCL utility and press ENTER.
Enter END command to cancel MODEL command.

J1 COMMAND - Command card
J2 DD - DD card
J3 JCLLIB - Proclib card
J4 JOB - Job card
J5 JOBPARM - Jobparm card
J6 IF - If card
J7 INCLUDE - Include card
J8 OUTPUT - Output card
J9 ROUTE - Route card
J10 SET - Set card

More: -
O29 SAVRS - Obtain syslog/reports
O30 SDEBYSDE - Side by side comparison
O31 SHOWMVS - IPL information
O32 SYSWAIT - Wait a duration of time
O33 TAPEMAP - Print contents of tape
O34 TASID - System Monitor
O35 TMSBINQ - Inquire on CA1 tape
O36 UTGZ - Send message to console
O37 UTJE - Abend on program RC
O38 UTJM - Empty sequential DSN
O39 UTJP - Issue MVS commands
O40 UTKT - Abend job w/ user code
O41 UTONY - Print PDS members
O42 VSPRIME - VSAM Prime

Application

A1 APPLJCL - Personal/Team JCL

JCL – ISREM JCL – Panel

```

)PANEL KEYLIST(ISRSNAB,ISR)
/*-----*/
/* This panel is invoked by using the TSO edit command "MODEL",
/* which in turn invokes panel ISREMCLS. Below is the flow-chart:
/*
/*
/*      MODEL  --> ISREMCLS  --> ISREMJCL  --> M*****
/*      IBM Cmd   IBM panel   User Panel   User Skeleton
/*
/*
/*  FLG  YYMMDD  USERID  DESCRIPTION
/*-----*/
/* @B2  120729  R2221  Added FTPSEC
/* @B1  120717  R2221  Added ICETOOL
/* @AZ  120629  R2221  Added PDSMERGE
/* @AY  120523  R2221  Added MZCCMPB1 and EXAMBATX
/* @AX  120113  R2221  Added AFCLOGR1 and FCSWAIT
/* @AW  120113  R2221  Added DAF
/* @AU  110602  R2221  Added VSPRIME
/* @AT  110518  R2221  Added PDSM36
/* @AS  110206  R2221  Added PDSMAN programs
/* @AR  100818  R2221  Added BPXBATCH (BPXBATSL)
/* @AQ  100812  R2221  Chged SYSAFF to JOBPARM
/* @AP  100812  R2221  Removed COMMAND Utility
/* @AO  100812  R2221  Added JCL COMMAND, IF, INCLUDE and SET
/* @AN  100809  R2221  Added COMPAREB and SDEBYSDE
/* @AM  100714  R2221  Removed Insync
/* @AL  100702  R2221  Added PDS
/* @AK  100630  R2221  Fixed Point-n-shoot
/* @AJ  100628  R2221  Added SYSWAIT
/* @AI  100611  R2221  Chged QW to QWIKREF1
/* @AH  100609  R2221  Added CAL2X2WB, CA7 commands using CCI
/* @AG  100513  R2221  Added UTGZ, UTJE/JM/JP, UTKT and UTONY
/* @AF  100507  R2221  Added FILEAID adm DJCTEST
/* @AE  100504  R2221  Added IRXJCL and REXXCL
/* @AD  100503  R2221  Fixed ICEMAN and QW when passing as parm
/* @AC  100503  R2221  Added Cobol to Extreme Converter
/* @AB  100425  R2221  Added capability for application JCL
/* @AA  100419  R2221  Removed NDM, suggest using NDMDSN utility
/* @A9  100126  R2221  Added CPUPDTE
/* @A8  091228  R2221  Added ROUTE card
/* @A7  091202  R2221  Added TASID, SHOWMVS, IFCEREPL, IFASMFDP
/* @A6  091006  R2221  Consolidated tape entries into OEM / Misc.
/* @A5  090922  R2221  Add INSYNC
/* @A4  090917  R2221  Add OUTPUT, SYSAFF, and JCLLIB
/* @A3  090916  R2221  Add AMATERSE, CONSOLE2, GENMAIL, and SAVRS
/* @A2  090915  R2221  Removed invalid software
/* @A1  011025  S915460  Added descriptions to programs
/* @A0  020225  S915460  Added FDRMAP
/*-----*/
)ATTR DEFAULT(%+_) FORMAT(MIX)          /* ISREMPLI - ENGLISH - 5.0 */
% TYPE(TEXT) INTENS(HIGH) COLOR(TURQ)

```

```

+ TYPE(TEXT) INTENS(LOW) COLOR(GREEN)
[ TYPE(PS)
] TYPE(NT)
0D TYPE(PS)
05 TYPE(PT)
06 TYPE(PIN)
09 TYPE(FF)
0A TYPE(NT)
11 TYPE(SAC)
22 TYPE(WASL) SKIP(ON) GE(ON)
10 TYPE(ET)
27 TYPE(NEF) CAPS(ON) PADC(USER)
28 AREA(SCRL) EXTEND(ON)
)BODY WINDOW(80,22) EXPAND(..) CMD(ZCMD)
.. .JCL Utility Models ..
.Option ==>.Z. .
.
.Enter an option or move the cursor on JCL utility and press ENTER.
.Enter.END.command to cancel MODEL command.
.SAREA40
)AREA SAREA40
%IBM %OEM+
+I1 [AMATERSE]- Terse a file
+I2 [AMBLIST]- Print linkage attributes
+I3 [BPXBATCHE]- Run USS shell scripts
+I4 [CPUPDTE]- Search/update
+I5 [DFDSS]- Backup/restore files
+I6 [DFSORT]- Sort/merge files
+I7 [ICETOOL]- Sort/merge files
+I8 [ICKDSF]- Init disk pack
+I9 [IDCAMS]- Def/del/alter VSAM/GDGs
+I10[IEBCOPY]- Copy/del/move PDS mbr
+I11[IEBGENER]- Copy sequential files
+I12[IEBPTPCH]- Print and punch
+I13[IEBUPDTE]- Create/alter PDS mbr
+I14[IEHLIST]- List PDS directory
+I15[IEHPRGDM]- Rename/del/cat file
+I16[IEFBRL14]- Create/delete file
+I17[IFASMFDP]- Dump SMF records
+I18[IFCEREPL]- Dump LOGREC
+I19[IKJEFT01]- TSO batch
+I20[IMASZPAP]- Zap/dump a module
+I21[IRXJCL]- Execute REXX program
+I22[ISRSUPC]- Search/compare
+I23[REXXCL]- Compile/link REXX pgm
+I24[TRSMMAIN]- Terse a file
+I25[SDSF]- SDSF batch
+I26[SMP]- IBM's ChangeMan
+
%JCL+

```


JCL – ISREM JCL – Panel

```
019,'PGM(ISRECMR) PARM(MPDSM13 )'  
020,'PGM(ISRECMR) PARM(MPDSM18 )'  
021,'PGM(ISRECMR) PARM(MPDSM25 )'  
022,'PGM(ISRECMR) PARM(MPDSM32 )'  
023,'PGM(ISRECMR) PARM(MPDSM33 )'  
024,'PGM(ISRECMR) PARM(MPDSM36 )'  
025,'PGM(ISRECMR) PARM(MPDSMERG)'  
026,'PGM(ISRECMR) PARM(MPKZIP )'  
027,'PGM(ISRECMR) PARM(MQWIKREF)'  
028,'PGM(ISRECMR) PARM(MSASSBST)'  
029,'PGM(ISRECMR) PARM(MSAVRS )'  
030,'PGM(ISRECMR) PARM(MSDEBYS)'  
031,'PGM(ISRECMR) PARM(MSHOWMVS)'  
032,'PGM(ISRECMR) PARM(MSYSWAIT)'  
033,'PGM(ISRECMR) PARM(MTAPEMAP)'  
034,'PGM(ISRECMR) PARM(MTASID )'  
035,'PGM(ISRECMR) PARM(MTMSBINQ)'  
036,'PGM(ISRECMR) PARM(MUTGZ )'  
037,'PGM(ISRECMR) PARM(MUTJE )'  
038,'PGM(ISRECMR) PARM(MUTJM )'  
039,'PGM(ISRECMR) PARM(MUTJP )'  
040,'PGM(ISRECMR) PARM(MUTKT )'  
041,'PGM(ISRECMR) PARM(MUTONY )'  
042,'PGM(ISRECMR) PARM(MVSPRIME)'  
J1,'PGM(ISRECMR) PARM(MCOMMAND)'  
J2,'PGM(ISRECMR) PARM(MDD )'  
J3,'PGM(ISRECMR) PARM(MJCLLIB )'  
J4,'PGM(ISRECMR) PARM(MJOB )'  
J5,'PGM(ISRECMR) PARM(MJOBPARM)'  
J6,'PGM(ISRECMR) PARM(MIF )'  
J7,'PGM(ISRECMR) PARM(MINCLUDE)'  
J8,'PGM(ISRECMR) PARM(MOUTPUT )'  
J9,'PGM(ISRECMR) PARM(MROUTE )'  
J10,'PGM(ISRECMR) PARM(MSET )'  
A1,'PANEL(ISREMAP)'  
' , ' '  
* , '?' )  
IF (&ZSEL = '?' )  
  &ZEMTEMP = &ZCMD  
  .MSG = ISRYM013  
  &ZCMD = ' '  
  IF (&ZMACRO = 'Y') /* If a macro command @OY38423*/  
    .RESP = 'END' /* don't display error @OY38423*/  
    &ISRMMEMB = '?' /* Indicate no member fnd@OY38423*/  
  &ISRMMEND = 'N' /* SET THE END INDICATOR TO NO */  
  IF (.RESP = END) /* IF ENDING, WHY ... WHO CAUSED */  
    IF (&ISRMONCL = 'Y') /* MAKE SURE ITS NOT A CLASS OP. */  
      IF (&ISRMDSP = 'RETURN ') /* MAKE SURE ITS NOT END ON MBR. */  
        &ISRMMEND = 'Y' /* NO - ITS BECAUSE USER HIT END */  
    &ZMSG = .MSG /* Set message variable */
```

```
VPUT(ZMSG) PROFILE /* Save message in prof pool */  
)PNTS  
FIELD(ZPS01001) VAR(ZCMD) VAL(I1)  
FIELD(ZPS01002) VAR(ZCMD) VAL(O1)  
FIELD(ZPS01003) VAR(ZCMD) VAL(I2)  
FIELD(ZPS01004) VAR(ZCMD) VAL(O2)  
FIELD(ZPS01005) VAR(ZCMD) VAL(I3)  
FIELD(ZPS01006) VAR(ZCMD) VAL(O3)  
FIELD(ZPS01007) VAR(ZCMD) VAL(I4)  
FIELD(ZPS01008) VAR(ZCMD) VAL(O4)  
FIELD(ZPS01009) VAR(ZCMD) VAL(I5)  
FIELD(ZPS01010) VAR(ZCMD) VAL(O5)  
FIELD(ZPS01011) VAR(ZCMD) VAL(I6)  
FIELD(ZPS01012) VAR(ZCMD) VAL(O6)  
FIELD(ZPS01013) VAR(ZCMD) VAL(I7)  
FIELD(ZPS01014) VAR(ZCMD) VAL(O7)  
FIELD(ZPS01015) VAR(ZCMD) VAL(I8)  
FIELD(ZPS01016) VAR(ZCMD) VAL(O8)  
FIELD(ZPS01017) VAR(ZCMD) VAL(I9)  
FIELD(ZPS01018) VAR(ZCMD) VAL(O9)  
FIELD(ZPS01019) VAR(ZCMD) VAL(I10)  
FIELD(ZPS01020) VAR(ZCMD) VAL(O10)  
FIELD(ZPS01021) VAR(ZCMD) VAL(I11)  
FIELD(ZPS01022) VAR(ZCMD) VAL(O11)  
FIELD(ZPS01023) VAR(ZCMD) VAL(I12)  
FIELD(ZPS01024) VAR(ZCMD) VAL(O12)  
FIELD(ZPS01025) VAR(ZCMD) VAL(I13)  
FIELD(ZPS01026) VAR(ZCMD) VAL(O13)  
FIELD(ZPS01027) VAR(ZCMD) VAL(I14)  
FIELD(ZPS01028) VAR(ZCMD) VAL(O14)  
FIELD(ZPS01029) VAR(ZCMD) VAL(I15)  
FIELD(ZPS01030) VAR(ZCMD) VAL(O15)  
FIELD(ZPS01031) VAR(ZCMD) VAL(I16)  
FIELD(ZPS01032) VAR(ZCMD) VAL(O16)  
FIELD(ZPS01033) VAR(ZCMD) VAL(I17)  
FIELD(ZPS01034) VAR(ZCMD) VAL(O17)  
FIELD(ZPS01035) VAR(ZCMD) VAL(I18)  
FIELD(ZPS01036) VAR(ZCMD) VAL(O18)  
FIELD(ZPS01037) VAR(ZCMD) VAL(I19)  
FIELD(ZPS01038) VAR(ZCMD) VAL(O19)  
FIELD(ZPS01039) VAR(ZCMD) VAL(I20)  
FIELD(ZPS01040) VAR(ZCMD) VAL(O20)  
FIELD(ZPS01041) VAR(ZCMD) VAL(I21)  
FIELD(ZPS01042) VAR(ZCMD) VAL(O21)  
FIELD(ZPS01043) VAR(ZCMD) VAL(I22)  
FIELD(ZPS01044) VAR(ZCMD) VAL(O22)  
FIELD(ZPS01045) VAR(ZCMD) VAL(I23)  
FIELD(ZPS01046) VAR(ZCMD) VAL(O23)  
FIELD(ZPS01047) VAR(ZCMD) VAL(I24)  
FIELD(ZPS01048) VAR(ZCMD) VAL(O24)
```

JCL – ISREMJCL – Panel

```
FIELD(ZPS01049) VAR(ZCMD) VAL(I25)
FIELD(ZPS01050) VAR(ZCMD) VAL(O25)
FIELD(ZPS01051) VAR(ZCMD) VAL(I26)
FIELD(ZPS01052) VAR(ZCMD) VAL(O26)
FIELD(ZPS01053) VAR(ZCMD) VAL(O27)
FIELD(ZPS01054) VAR(ZCMD) VAL(O28)
FIELD(ZPS01055) VAR(ZCMD) VAL(J1)
FIELD(ZPS01056) VAR(ZCMD) VAL(O29)
FIELD(ZPS01057) VAR(ZCMD) VAL(J2)
FIELD(ZPS01058) VAR(ZCMD) VAL(O30)
FIELD(ZPS01059) VAR(ZCMD) VAL(J3)
FIELD(ZPS01060) VAR(ZCMD) VAL(O31)
FIELD(ZPS01061) VAR(ZCMD) VAL(J4)
FIELD(ZPS01062) VAR(ZCMD) VAL(O32)
FIELD(ZPS01063) VAR(ZCMD) VAL(J5)
FIELD(ZPS01064) VAR(ZCMD) VAL(O33)
FIELD(ZPS01065) VAR(ZCMD) VAL(J6)
FIELD(ZPS01066) VAR(ZCMD) VAL(O34)
FIELD(ZPS01067) VAR(ZCMD) VAL(J7)
FIELD(ZPS01068) VAR(ZCMD) VAL(O35)
FIELD(ZPS01069) VAR(ZCMD) VAL(J8)
FIELD(ZPS01070) VAR(ZCMD) VAL(O36)
FIELD(ZPS01071) VAR(ZCMD) VAL(J9)
FIELD(ZPS01072) VAR(ZCMD) VAL(O37)
FIELD(ZPS01073) VAR(ZCMD) VAL(J10)
FIELD(ZPS01074) VAR(ZCMD) VAL(O38)
FIELD(ZPS01075) VAR(ZCMD) VAL(O39)
FIELD(ZPS01076) VAR(ZCMD) VAL(O40)
FIELD(ZPS01077) VAR(ZCMD) VAL(O41)
FIELD(ZPS01078) VAR(ZCMD) VAL(O42)
FIELD(ZPS01079) VAR(ZCMD) VAL(A1)
)END
```

JCL – Sample Skeletons

MIEHLIST

```
/*-----*
/*  PURPOSE:  PDS, VTOC OR CATALOG LISTING
/*-----*
/*  MANUAL:   DFSMSDFP UTILITIES
/*-----*
//IEHLIST EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=*
//DDL1 DD DISP=OLD,UNIT=SYSALLDA,VOL=SER=#####
//SYSIN DD *
)N*-----*
)N* List a PDS directory in hexadecimal format
)N*-----*
)N* LISTPDS DSNAME=(*****.*****.*****), X
)N* VOL=SYSDA=#####
)N*-----*
)N* List a PDS directory attributes (only valid for executables)
)N*-----*
)N* LISTPDS DSNAME=(*****.*****.*****),FORMAT, X
)N* VOL=SYSDA=#####
)N*-----*
)N* Retrieve a VTOC listing
)N*-----*
)N* LISTVTOC FORMAT,VOL=DISK=#####
)N*-----*
)N* Partitioned Dataset (PDS) listing
)N*-----*
)N* LISTPDS {DSNAME=(dsn1{,dsn2...})}
)N* {,DUMP|FORMAT}
)N* {,VOL=device=serial}
)N*-----*
)N* Volume Table of Contents (VTOC) listing
)N*-----*
)N* LISTVTOC {DUMP|FORMAT}
)N* {,INDEXDSN=SYS1.VTOCIX.xxxxxx}
)N* {,DATE=ddyy}
)N* {,VOL=device=serial}
)N* {,DSNAME=(dsn1{,dsn2...})}
)N*-----*
)N* Catalog Listing
)N*-----*
)N* LISTCTLG {VOL=device=serial}
)N* {,NODE=name}
)N
```

MUTKT

```
/*-----*
/*  PURPOSE:  ABEND JOB WITH USER CODE
/*-----*
//STEP010 EXEC PGM=UTKT,PARM=999,COND=(0,EQ,?????)
)N*-----*
)N* Description:
)N* 'UTKT' was written to provide a graceful abend exit
)N* point for test and production jobs when JCL execution
)N* needs to be stopped due to prior step execution return
)N* codes.
)N*-----*
)N* PARM field:
)N* The number of characters present in the PARM field
)N* determines the user abend code to be issued. In the
)N* below example, UTKT abends with a user 0003.
)N*-----*
)N* PARM field limitations:
)N* - Can contain any combination of alphanumeric and
)N* national ($,#,@) characters.
)N* - Up to 100 characters are allowed
)N*-----*
```

ISREMRXC – Panel

REXX Models

Option ==>

Enter a number or move the cursor to a line and press Enter.
Enter the END command to cancel the MODEL command.

SDSF REXX
S0 SDSFREXX

ISPF Workstation

X1 FILESTAT
X2 FILEXFER
X3 WSCON
X4 WSDISCON

ISPF Library Access

L0 LIBACC

ISPF Variables

V1 VGET
V2 VPUT
V3 VERASE
V4 VSYM

ISPF Miscellaneous

M0 MISC

ISPF File Tailoring

F1 FTOPEN
F2 FTINCL
F3 FTCLOSE
F4 FTERASE

REXX Commands

RX REXXCMS

ISPF Display Services

D1 DISPLAY
D2 TBDISPL
D3 SETMSG
D4 PQUERY
D5 ADDPOP
D6 REMPOP

ISPF Tables

T1 TABLES

SDSF REXX



- IBM System Display and Search Facility Downloads
 - <http://www-03.ibm.com/systems/z/os/zos/features/sdsf/tools/sdsfrexx.html>

REXX – Commands – Implementation

- Create ISRRXCMD panel
- Invoke MVS/Quick Reference (**QW**) and select option
 - “**C**”, Request information by Category
 - “PROGLANG”, Programming languages syntax
 - “z/OS REXX SYNTAX”
 - A REXX command
 - Type in “**CUT ALL**”
- Create a new skeleton member
 - Type in **QWPASTE**
 - Shift the text over three characters to the right
 - Insert “**)N**” to make all lines notes
- Add/update an entry to REXX model panel (**ISRRXCMD**)

REXX – ISRRXCMD – Panel

REXX Commands and Functions

Option ===>

Enter a REXX command or move the cursor over REXX command and press ENTER.
Enter END command to cancel MODEL command.

More: +

ABBREV	ABS	ADDRESS	ARG	BITAND	BITOR
BITXOR	BOOLEAN	B2X	CALL	CENTER	COMPARE
CONDITION	COPIES	C2D	C2X	DATATYPE	DATE
DELSTACK	DELSTR	DELWORD	DIGITS	DO	DROP
DROPBUF	D2C	D2X	ERRORTEXT	EXECIO	EXECUTIL
EXIT	EXTERNALS	FIND	FORM	FORMAT	FUZZ
GETMSG	HE	HI	HT	IF	INDEX
INSERT	INTERPRET	ITERATE	JUSTIFY	LASTPOS	LEAVE
LEFT	LENGTH	LINESIZE	LISTDSI	MAKEBUF	MAX
MIN	MSG	MVSVAR	NEWSTACK	NOP	NUMERIC
OPTIONS	OUTTRAP	OVERLAY	PARSE	POS	PROCEDURE
PROMPT	PULL	PUSH	QBUF	QELEM	QSTACK
QUEUE	QUEUED	RANDOM	RETURN	REVERSE	RIGHT
RT	SAY	SELECT	SETLANG	SIGN	SIGNAL
SOURCELINE	SPACE	STORAGE	STRIP	SUBCOM	SUBSTR
SUBWORD	SYMBOL	SYSCPUS	SYSDSN	SYSVAR	TE
TIME	TRACE	TRANSLATE	TRUNC	TS	UNTIL
UPPER	USERID	VALUE	VERIFY	WHILE	WORD

REXX – ISRRXCMD – Panel

```

)PANEL KEYLIST(ISRSNAB,ISR)
/*-----*/
/* This panel is invoked by using the TSO edit command "MODEL", */
/* which in turn invokes panel ISREMCLS. Below is the flow-chart: */
/* */
/* MODEL --> ISREMRXC --> ISRRXCMD --> M***** */
/* IBM Cmd IBM panel User Panel User Skeleton */
/* */
/* FLG YMMDD USERID DESCRIPTION */
/*-----*/
/* @A0 100630 R2221 Developed Panel */
/*-----*/
)ATTR DEFAULT(%+_) FORMAT(MIX) /* ISREMPLE - ENGLISH - 5.0 */
[ TYPE(PS)
] TYPE(NT)
0D TYPE(PS)
05 TYPE(PT)
06 TYPE(PIN)
09 TYPE(FP)
0A TYPE(NT)
11 TYPE(SAC)
22 TYPE(WASL) SKIP(ON) GE(ON)
10 TYPE(ET)
27 TYPE(NEF) CAPS(ON) PADC(USER)
28 AREA(SCRL) EXTEND(ON)
)BODY WINDOW(80,22) EXPAND(..) CMD(ZCMD)
.. .REXX Commands and Functions. ..
.Option ==>.Z. .
.
.Enter a REXX command or move the cursor over REXX command and press ENTER.
.Enter.END.command to cancel MODEL command.
.SAREA40
)AREA SAREA40
+ [ABBREV ][ABS ][ADDRESS ][ARG ][BITAND ][BITOR ]
+ [BITXOR ][BOOLEAN ][B2X ][CALL ][CENTER ][COMPARE ]
+ [CONDITION ][COPIES ][C2D ][C2X ][DATATYPE][DATE ]
+ [DELSTACK ][DELSTR ][DELWORD ][DIGITS ][DO ][DROP ]
+ [DROPEBUF ][D2C ][D2X ][ERRORTEXT][EXECIO ][EXECUTIL]
+ [EXIT ][EXTERNALS ][FIND ][FORM ][FORMAT ][FUZZ ]
+ [GETMSG ][HE ][HI ][HT ][IF ][INDEX ]
+ [INSERT ][INTERPRET ][ITERATE ][JUSTIFY ][LASTPOS ][LEAVE ]
+ [LEFT ][LENGTH ][LINESIZE ][LISTDSI ][MAKEBUF ][MAX ]
+ [MIN ][MSG ][MVSVAR ][NEWSTACK ][NOP ][NUMERIC ]
+ [OPTIONS ][OUTTRAP ][OVERLAY ][PARSE ][POS ][PROCEDURE
+ [PROMPT ][PULL ][PUSH ][QBUF ][QLEM ][QSTACK ]
+ [QUEUE ][QUEUEUED ][RANDOM ][RETURN ][REVERSE ][RIGHT ]
+ [RT ][SAY ][SELECT ][SETLANG ][SIGN ][SIGNAL ]
+ [SOURCELINE][SPACE ][STORAGE ][STRIP ][SUBCOM ][SUBSTR ]
+ [SUBWORD ][SYMBOL ][SYSCPUS ][SYSDSN ][SYSVAR ][TE ]
+ [TIME ][TRACE ][TRANSLATE][TRUNC ][TS ][UNTIL ]

```

```

+ [UPPER ][USERID ][VALUE ][VERIFY ][WHILE ][WORD ]
+ [WORDINDEX ][WORDLENGTH][WORDPOS ][WORDS ][XRANGE ][X2B ]
+ [X2C ][X2D ]
)INIT
.ZVARS = '(ZCMD)'
&ZWINTTL = ' '
.HELP = #ISREMJC
IF (&ISRMSPL = 'RETURN ')
.RESP = END
.CURSOR = 'ZCMD'
)REINIT
REFRESH(ZCMD)
)PROC
&ZCMD = TRUNC (&ZCMD, '.')
&ZTMPTRL = .TRAIL
IF (&ZCMD ^= ' ') &ZCMD = '&ZCMD..&ZTMPTRL'
&ZSEL = TRANS (TRUNC (&ZCMD, '.'))
ABBREV , 'PGM(ISRECMBR) PARM(MRXABBRE)'
ABS , 'PGM(ISRECMBR) PARM(MRXABS)'
ADDRESS , 'PGM(ISRECMBR) PARM(MRXADDRE)'
ARG , 'PGM(ISRECMBR) PARM(MRXARG)'
BITAND , 'PGM(ISRECMBR) PARM(MRXBITAN)'
BITOR , 'PGM(ISRECMBR) PARM(MRXBITOR)'
BITXOR , 'PGM(ISRECMBR) PARM(MRXBITXO)'
BOOLEAN , 'PGM(ISRECMBR) PARM(MRXBOOLE)'
B2X , 'PGM(ISRECMBR) PARM(MRXB2X)'
CALL , 'PGM(ISRECMBR) PARM(MRXCALL)'
CENTER , 'PGM(ISRECMBR) PARM(MRXCENTE)'
COMPARE , 'PGM(ISRECMBR) PARM(MRXCOMPA)'
CONDITION , 'PGM(ISRECMBR) PARM(MRXCONDI)'
COPIES , 'PGM(ISRECMBR) PARM(MRXCOPIE)'
C2D , 'PGM(ISRECMBR) PARM(MRXC2D)'
C2X , 'PGM(ISRECMBR) PARM(MRXC2X)'
DATATYPE , 'PGM(ISRECMBR) PARM(MRXDATAT)'
DATE , 'PGM(ISRECMBR) PARM(MRXDATE)'
DELSTACK , 'PGM(ISRECMBR) PARM(MRXDELS)'
DELSTR , 'PGM(ISRECMBR) PARM(MRXDELST)'
DELWORD , 'PGM(ISRECMBR) PARM(MRXDELWO)'
DIGITS , 'PGM(ISRECMBR) PARM(MRXDIGIT)'
DO , 'PGM(ISRECMBR) PARM(MRXDO)'
DROP , 'PGM(ISRECMBR) PARM(MRXDROP)'
DROPEBUF , 'PGM(ISRECMBR) PARM(MRXDROPB)'
D2C , 'PGM(ISRECMBR) PARM(MRXD2C)'
D2X , 'PGM(ISRECMBR) PARM(MRXD2X)'
ERRORTEXT , 'PGM(ISRECMBR) PARM(MRXERROR)'
EXECIO , 'PGM(ISRECMBR) PARM(MRXEXECIO)'
EXECUTIL , 'PGM(ISRECMBR) PARM(MRXEXECU)'
EXIT , 'PGM(ISRECMBR) PARM(MRXEXIT)'
EXTERNALS , 'PGM(ISRECMBR) PARM(MRXEXTER)'
FIND , 'PGM(ISRECMBR) PARM(MRXFIND)'

```

REXX – ISRRXCMD – Panel

```

FORM          , 'PGM(ISRECMR) PARM(MRXFORM) '
FORMAT        , 'PGM(ISRECMR) PARM(MRXFORMA)'
FUZZ          , 'PGM(ISRECMR) PARM(MRXFUZZ) '
GETMSG        , 'PGM(ISRECMR) PARM(MRXGETMS)'
HE            , 'PGM(ISRECMR) PARM(MRXHE)   '
HI            , 'PGM(ISRECMR) PARM(MRXHI)   '
HT            , 'PGM(ISRECMR) PARM(MRXHT)   '
IF            , 'PGM(ISRECMR) PARM(MRXIF)   '
INDEX         , 'PGM(ISRECMR) PARM(MRXINDEX)'
INSERT        , 'PGM(ISRECMR) PARM(MRXINSER)'
INTERPRET     , 'PGM(ISRECMR) PARM(MRXINTER)'
ITERATE       , 'PGM(ISRECMR) PARM(MRXITERA)'
JUSTIFY       , 'PGM(ISRECMR) PARM(MRXJUSTI)'
LASTPOS       , 'PGM(ISRECMR) PARM(MRXLASTP)'
LEAVE         , 'PGM(ISRECMR) PARM(MRXLEAVE)'
LEFT          , 'PGM(ISRECMR) PARM(MRXLEFT) '
LENGTH        , 'PGM(ISRECMR) PARM(MRXLENGT)'
LINESIZE      , 'PGM(ISRECMR) PARM(MRXLINES)'
LISTDSI       , 'PGM(ISRECMR) PARM(MRXLISTD)'
MAKEBUF       , 'PGM(ISRECMR) PARM(MRXMAKEB)'
MAX           , 'PGM(ISRECMR) PARM(MRXMAX)   '
MIN           , 'PGM(ISRECMR) PARM(MRXMIN)   '
MSG           , 'PGM(ISRECMR) PARM(MRXMSG)   '
MVSVAR        , 'PGM(ISRECMR) PARM(MRXMVSVA)'
NEWSTACK      , 'PGM(ISRECMR) PARM(MRXNEWST)'
NOP           , 'PGM(ISRECMR) PARM(MRXNOP)   '
NUMERIC       , 'PGM(ISRECMR) PARM(MRXNUMBER)'
OPTIONS       , 'PGM(ISRECMR) PARM(MRXOPTIO)'
OUTTRAP       , 'PGM(ISRECMR) PARM(MRXOUTTR)'
OVERLAY       , 'PGM(ISRECMR) PARM(MRXOVERL)'
PARSE         , 'PGM(ISRECMR) PARM(MRXPARSE)'
POS           , 'PGM(ISRECMR) PARM(MRXPOS)   '
PROCEDURE     , 'PGM(ISRECMR) PARM(MRXPROC)'
PROMPT        , 'PGM(ISRECMR) PARM(MRXPROMP)'
PULL          , 'PGM(ISRECMR) PARM(MRXPULL) '
PUSH          , 'PGM(ISRECMR) PARM(MRXPUSH) '
QBUF          , 'PGM(ISRECMR) PARM(MRXQBUF) '
QELEM         , 'PGM(ISRECMR) PARM(MRXQELEM)'
QSTACK        , 'PGM(ISRECMR) PARM(MRXQSTAC)'
QUEUE         , 'PGM(ISRECMR) PARM(MRXQUEUE)'
QUEUED        , 'PGM(ISRECMR) PARM(MRXQUEUD)'
RANDOM         , 'PGM(ISRECMR) PARM(MRXRANDO)'
RETURN        , 'PGM(ISRECMR) PARM(MRXRETUR)'
REVERSE       , 'PGM(ISRECMR) PARM(MRXREVER)'
RIGHT         , 'PGM(ISRECMR) PARM(MRXRIGHT)'
RT            , 'PGM(ISRECMR) PARM(MRXRT)   '
SAY           , 'PGM(ISRECMR) PARM(MRXSAY)   '
SELECT        , 'PGM(ISRECMR) PARM(MRXSELEC)'
SETLANG       , 'PGM(ISRECMR) PARM(MRXSETLA)'
SIGN          , 'PGM(ISRECMR) PARM(MRXSIGN) '
    
```

```

SIGNAL        , 'PGM(ISRECMR) PARM(MRXSIGNA)'
SOURCELINE    , 'PGM(ISRECMR) PARM(MRXSOURC)'
SPACE         , 'PGM(ISRECMR) PARM(MRXSPACE)'
STORAGE       , 'PGM(ISRECMR) PARM(MRXSTORA)'
STRIP         , 'PGM(ISRECMR) PARM(MRXSTRIP)'
SUBCOM        , 'PGM(ISRECMR) PARM(MRXSUBCO)'
SUBSTR        , 'PGM(ISRECMR) PARM(MRXSUBST)'
SUBWORD       , 'PGM(ISRECMR) PARM(MRXSUBWO)'
SYMBOL        , 'PGM(ISRECMR) PARM(MRXSYMBO)'
SYSCPUS       , 'PGM(ISRECMR) PARM(MRXSYSCP)'
SYSDSN        , 'PGM(ISRECMR) PARM(MRXSYSDS)'
SYSVAR        , 'PGM(ISRECMR) PARM(MRXSYSVA)'
TE            , 'PGM(ISRECMR) PARM(MRXTE)   '
TIME          , 'PGM(ISRECMR) PARM(MRXTIME) '
TRACE         , 'PGM(ISRECMR) PARM(MRXTRACE)'
TRANSLATE     , 'PGM(ISRECMR) PARM(MRXTRANS)'
TRUNC         , 'PGM(ISRECMR) PARM(MRXTRUNC)'
TS            , 'PGM(ISRECMR) PARM(MRXTS)   '
UNTIL         , 'PGM(ISRECMR) PARM(MRXUNTIL)'
UPPER         , 'PGM(ISRECMR) PARM(MRXUPPER)'
USERID        , 'PGM(ISRECMR) PARM(MRXUSERI)'
VALUE         , 'PGM(ISRECMR) PARM(MRXVALUE)'
VERIFY        , 'PGM(ISRECMR) PARM(MRXVERIF)'
WHILE         , 'PGM(ISRECMR) PARM(MRXWHILE)'
WORD          , 'PGM(ISRECMR) PARM(MRXWORD) '
WORDINDEX     , 'PGM(ISRECMR) PARM(MRXWORDI)'
WORDLENGTH    , 'PGM(ISRECMR) PARM(MRXWORDL)'
WORDPOS       , 'PGM(ISRECMR) PARM(MRXWORDP)'
WORDS         , 'PGM(ISRECMR) PARM(MRXWORDS)'
XRANGE        , 'PGM(ISRECMR) PARM(MRXXRANG)'
X2B           , 'PGM(ISRECMR) PARM(MRX2B)   '
X2C           , 'PGM(ISRECMR) PARM(MRX2C)   '
X2D           , 'PGM(ISRECMR) PARM(MRX2D)   '
' , ' , ' , '
' , '?' '
IF (&ZSEL = '?' )
  &ZEMTEMP = &ZCMD
  .MSG = ISRYM013
  &ZCMD = ' '
  *DLSLIST
IF (&ZMACRO = 'Y') /* If a macro command @OY38423*/
  .RESP = 'END' /* don't display error @OY38423*/
  &ISRMMEMB = '?' /* Indicate no member fnd@OY38423*/
&ISRMMEND = 'N' /* SET THE END INDICATOR TO NO */
IF (.RESP = END) /* IF ENDING, WHY ... WHO CAUSED */
  IF (&ISRMONCL = 'Y') /* MAKE SURE ITS NOT A CLASS OP. */
  IF (&ISRMSDPL = 'RETURN ') /* MAKE SURE ITS NOT END ON MBR. */
  &ISRMMEND = 'Y' /* NO - ITS BECAUSE USER HIT END */
&ZMSG = .MSG /* Set message variable */
VPUT(ZMSG) PROFILE /* Save message in prof pool */
    
```

REXX – ISRRXCMD – Panel

```
)PNTS  
FIELD(ZPS01001) VAR(ZCMD) VAL(ABBREV )  
FIELD(ZPS01002) VAR(ZCMD) VAL(ABS )  
FIELD(ZPS01003) VAR(ZCMD) VAL(ADDRESS )  
FIELD(ZPS01004) VAR(ZCMD) VAL(ARG )  
FIELD(ZPS01005) VAR(ZCMD) VAL(BITAND )  
FIELD(ZPS01006) VAR(ZCMD) VAL(BITOR )  
FIELD(ZPS01007) VAR(ZCMD) VAL(BITXOR )  
FIELD(ZPS01008) VAR(ZCMD) VAL(BOOLEAN )  
FIELD(ZPS01009) VAR(ZCMD) VAL(B2X )  
FIELD(ZPS01010) VAR(ZCMD) VAL(CALL )  
FIELD(ZPS01011) VAR(ZCMD) VAL(CENTER )  
FIELD(ZPS01012) VAR(ZCMD) VAL(COMPARE )  
FIELD(ZPS01013) VAR(ZCMD) VAL(CONDITION )  
FIELD(ZPS01014) VAR(ZCMD) VAL(COPIES )  
FIELD(ZPS01015) VAR(ZCMD) VAL(C2D )  
FIELD(ZPS01016) VAR(ZCMD) VAL(C2X )  
FIELD(ZPS01017) VAR(ZCMD) VAL(DATATYPE )  
FIELD(ZPS01018) VAR(ZCMD) VAL(DATE )  
FIELD(ZPS01019) VAR(ZCMD) VAL(DELSTACK )  
FIELD(ZPS01020) VAR(ZCMD) VAL(DELSTR )  
FIELD(ZPS01021) VAR(ZCMD) VAL(DELWORD )  
FIELD(ZPS01022) VAR(ZCMD) VAL(DIGITS )  
FIELD(ZPS01023) VAR(ZCMD) VAL(DO )  
FIELD(ZPS01024) VAR(ZCMD) VAL(DROP )  
FIELD(ZPS01025) VAR(ZCMD) VAL(DROPBUF )  
FIELD(ZPS01026) VAR(ZCMD) VAL(D2C )  
FIELD(ZPS01027) VAR(ZCMD) VAL(D2X )  
FIELD(ZPS01028) VAR(ZCMD) VAL(ERRORTEXT )  
FIELD(ZPS01029) VAR(ZCMD) VAL(EXECIO )  
FIELD(ZPS01030) VAR(ZCMD) VAL(EXECUTIL )  
FIELD(ZPS01031) VAR(ZCMD) VAL(EXIT )  
FIELD(ZPS01032) VAR(ZCMD) VAL(EXTERNALS )  
FIELD(ZPS01033) VAR(ZCMD) VAL(FIND )  
FIELD(ZPS01034) VAR(ZCMD) VAL(FORM )  
FIELD(ZPS01035) VAR(ZCMD) VAL(FORMAT )  
FIELD(ZPS01036) VAR(ZCMD) VAL(FUZZ )  
FIELD(ZPS01037) VAR(ZCMD) VAL(GETMSG )  
FIELD(ZPS01038) VAR(ZCMD) VAL(HE )  
FIELD(ZPS01039) VAR(ZCMD) VAL(HI )  
FIELD(ZPS01040) VAR(ZCMD) VAL(HT )  
FIELD(ZPS01041) VAR(ZCMD) VAL(IF )  
FIELD(ZPS01042) VAR(ZCMD) VAL(INDEX )  
FIELD(ZPS01043) VAR(ZCMD) VAL(INSERT )  
FIELD(ZPS01044) VAR(ZCMD) VAL(INTERPRET )  
FIELD(ZPS01045) VAR(ZCMD) VAL(ITERATE )  
FIELD(ZPS01046) VAR(ZCMD) VAL(JUSTIFY )  
FIELD(ZPS01047) VAR(ZCMD) VAL(LASTPOS )  
FIELD(ZPS01048) VAR(ZCMD) VAL(LEAVE )  
FIELD(ZPS01049) VAR(ZCMD) VAL(LEFT )
```

```
FIELD(ZPS01050) VAR(ZCMD) VAL(LENGTH )  
FIELD(ZPS01051) VAR(ZCMD) VAL(LINESIZE )  
FIELD(ZPS01052) VAR(ZCMD) VAL(LISTDSI )  
FIELD(ZPS01053) VAR(ZCMD) VAL(MAKEBUF )  
FIELD(ZPS01054) VAR(ZCMD) VAL(MAX )  
FIELD(ZPS01055) VAR(ZCMD) VAL(MIN )  
FIELD(ZPS01056) VAR(ZCMD) VAL(MSG )  
FIELD(ZPS01057) VAR(ZCMD) VAL(MVSVAR )  
FIELD(ZPS01058) VAR(ZCMD) VAL(NEWSTACK )  
FIELD(ZPS01059) VAR(ZCMD) VAL(NOP )  
FIELD(ZPS01060) VAR(ZCMD) VAL(NUMERIC )  
FIELD(ZPS01061) VAR(ZCMD) VAL(OPTIONS )  
FIELD(ZPS01062) VAR(ZCMD) VAL(OUTTRAP )  
FIELD(ZPS01063) VAR(ZCMD) VAL(OVERLAY )  
FIELD(ZPS01064) VAR(ZCMD) VAL(PARSE )  
FIELD(ZPS01065) VAR(ZCMD) VAL(POS )  
FIELD(ZPS01066) VAR(ZCMD) VAL(PROCEDURE )  
FIELD(ZPS01067) VAR(ZCMD) VAL(PROMPT )  
FIELD(ZPS01068) VAR(ZCMD) VAL(PULL )  
FIELD(ZPS01069) VAR(ZCMD) VAL(PUSH )  
FIELD(ZPS01070) VAR(ZCMD) VAL(QBUF )  
FIELD(ZPS01071) VAR(ZCMD) VAL(QELEM )  
FIELD(ZPS01072) VAR(ZCMD) VAL(QSTACK )  
FIELD(ZPS01073) VAR(ZCMD) VAL(Queue )  
FIELD(ZPS01074) VAR(ZCMD) VAL(QUEUED )  
FIELD(ZPS01075) VAR(ZCMD) VAL(RANDOM )  
FIELD(ZPS01076) VAR(ZCMD) VAL(RETURN )  
FIELD(ZPS01077) VAR(ZCMD) VAL(REVERSE )  
FIELD(ZPS01078) VAR(ZCMD) VAL(RIGHT )  
FIELD(ZPS01079) VAR(ZCMD) VAL(RT )  
FIELD(ZPS01080) VAR(ZCMD) VAL(SAY )  
FIELD(ZPS01081) VAR(ZCMD) VAL(SELECT )  
FIELD(ZPS01082) VAR(ZCMD) VAL(SETLANG )  
FIELD(ZPS01083) VAR(ZCMD) VAL(SIGN )  
FIELD(ZPS01084) VAR(ZCMD) VAL(SIGNAL )  
FIELD(ZPS01085) VAR(ZCMD) VAL(SOURCELINE )  
FIELD(ZPS01086) VAR(ZCMD) VAL(SPACE )  
FIELD(ZPS01087) VAR(ZCMD) VAL(STORAGE )  
FIELD(ZPS01088) VAR(ZCMD) VAL(STRIP )  
FIELD(ZPS01089) VAR(ZCMD) VAL(SUBCOM )  
FIELD(ZPS01090) VAR(ZCMD) VAL(SUBSTR )  
FIELD(ZPS01091) VAR(ZCMD) VAL(SUBWORD )  
FIELD(ZPS01092) VAR(ZCMD) VAL(SYMBOL )  
FIELD(ZPS01093) VAR(ZCMD) VAL(SYSCPUS )  
FIELD(ZPS01094) VAR(ZCMD) VAL(SYSDSN )  
FIELD(ZPS01095) VAR(ZCMD) VAL(SYSVAR )  
FIELD(ZPS01096) VAR(ZCMD) VAL(TE )  
FIELD(ZPS01097) VAR(ZCMD) VAL(TIME )  
FIELD(ZPS01098) VAR(ZCMD) VAL(TRACE )  
FIELD(ZPS01099) VAR(ZCMD) VAL(TRANSLATE )
```

REXX – ISRRXCMD – Panel

```
FIELD(ZPS01100) VAR(ZCMD) VAL(TRUNC )
FIELD(ZPS01101) VAR(ZCMD) VAL(TS )
FIELD(ZPS01102) VAR(ZCMD) VAL(UNTIL )
FIELD(ZPS01103) VAR(ZCMD) VAL(UPPER )
FIELD(ZPS01104) VAR(ZCMD) VAL(USERID )
FIELD(ZPS01105) VAR(ZCMD) VAL(VALUE )
FIELD(ZPS01106) VAR(ZCMD) VAL(VERIFY )
FIELD(ZPS01107) VAR(ZCMD) VAL(WHILE )
FIELD(ZPS01108) VAR(ZCMD) VAL(WORD )
FIELD(ZPS01109) VAR(ZCMD) VAL(WORDINDEX )
FIELD(ZPS01110) VAR(ZCMD) VAL(WORDLENGTH)
FIELD(ZPS01111) VAR(ZCMD) VAL(WORDPOS )
FIELD(ZPS01112) VAR(ZCMD) VAL(WORDS )
FIELD(ZPS01113) VAR(ZCMD) VAL(XRANGE )
FIELD(ZPS01114) VAR(ZCMD) VAL(X2B )
FIELD(ZPS01115) VAR(ZCMD) VAL(X2C )
FIELD(ZPS01116) VAR(ZCMD) VAL(X2D )
)END
```

REXX – Sample Skeletons

MRXPOS

```
)NPOS (Position)
)N
)N+-----|
)N|
)N| >>--POS(needle,haystack-----)>>|
)N|           +- ,start-+
)N|
)N+-----|
)N
)NReturns the position of one string, needle, in another haystack.
)N(See also the INDEX and LASTPOS functions.) Returns 0 if needle is
)Nthe null string or is not found or if start is greater than the
)Nlength of haystack. By default the search starts at the first
)Ncharacter of haystack (that is, the value of start is 1). You can
)Noverride this by specifying start (which must be a positive whole
)Nnumber), the point at which the search starts.
)N
)NHere are some examples:
)N
)N POS('day','Saturday')      -> 6
)N POS('x','abc def ghi')     -> 0
)N POS(' ','abc def ghi')     -> 4
)N POS(' ','abc def ghi',5)   -> 8
```

MRXSAY

```
)NSAY
)N
)N+-----|
)N|
)N| >>--SAY-----;----->>|
)N|           +-expression-+
)N|
)N+-----|
)N
)NSAY writes a line to the output stream. This typically displays it
)Nto the user, but the output destination can depend on the
)Nimplementation. The result of expression may be of any length. If
)Nyou omit expression, the null string is written.
)N
)NIf a REXX exec runs in TSO/E foreground, SAY displays the expression
)Nnon the terminal. The result from the SAY instruction is formatted
)Nto the current terminal line width (as defined by the TSO/E TERMINAL
)Ncommand) minus 1 character. In TSO/E background, SAY writes the
)Nexpression to the output stream, which is SYSTSPRT. In either case,
)Nwhen the length is undefined (LINESIZE() returns 0), SAY uses a
)Ndefault line size of 80.
)N
)NIf an exec runs in a non-TSO/E address space, SAY writes the
)Nexpression to the output stream as defined by the OUTDD field in the
)Nmodule name table (see item -MODNAMET). The system default is
)NSYSTSPRT. The ddname may be changed on an application basis or on a
)Nsystem basis.
)N
)NExample:
)N
)N data=100
)N say data 'divided by 4 =>' data/4
)N /* Displays: "100 divided by 4 => 25" */
```


Work Station Agent

Introduction



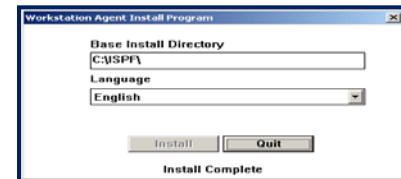
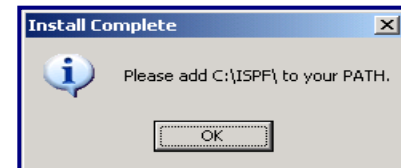
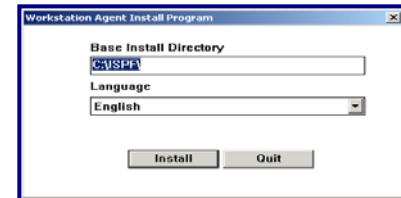
- Work Station Agent was introduced in ISPF V4 in MVS/ESA (1994)
- Customers finding WSA useful tend to
 - Have a lot of screen real-estate, several monitors connected to their PC
 - Prefer to use one editor to modify and view mainframe / PC files
 - Consistently download/upload files from their workstation
- Links
 - TSO Times, Spring 2004, Dave Salt
 - [Performing File Transfer using ISPF Work Station Agent \(WSA\)](#)
 - Appendix A, Installing the Client/Server Component
 - [ISPF – V1R13 – Users Guide – Vol 1](#)

Pros & Cons

- Pros
 - Easily transfer all members in a PDS file to a folder
 - SPLIT adds a new logical screen, acts like START
- Cons
 - 3270 emulator must be configured to use
 - Display Model = Model 2 - 24 rows 80 chrs
 - Maximum fixed window size, can not enlarge
 - Window can be resized smaller, but hides screen
 - Work Station Agent must be active
 - Not supported
 - Character-level color, intensity, and highlighting
 - Field-level intensity and highlighting
 - **FIND** does not work in browse mode
 - Can not initiate GUI display while in split screen mode

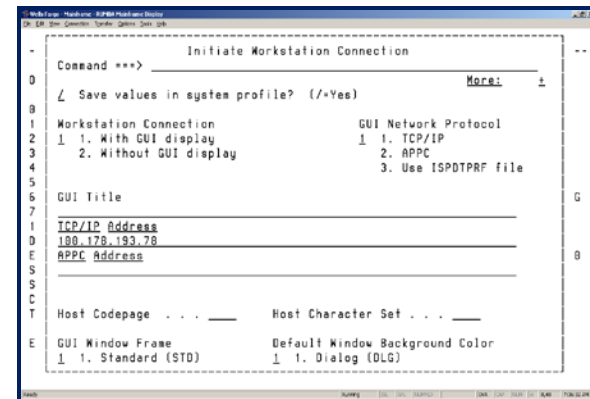
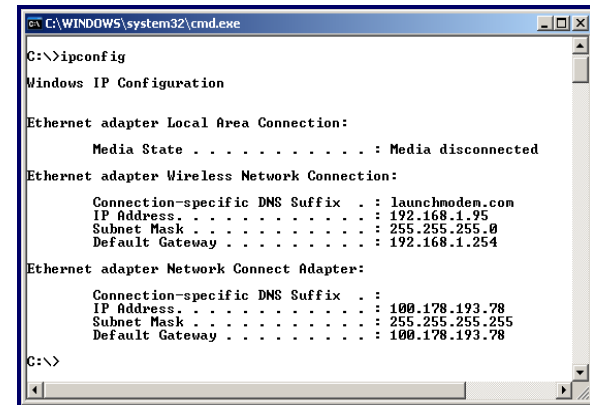
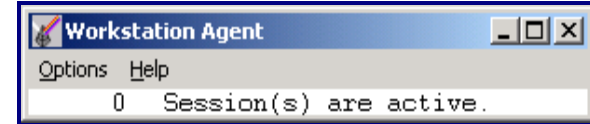
Install on PC

- Download below PDS member into Windows folder called “C:\ISPF”
 - [SYS1.SISPGUI\(ISPGUINX\)](#)
- Double click on “C:\ISPF\ISPGUINX.EXE”
 - Type in “C:\ISPF\” and click “Install”
 - Click “OK” button and disregard message
 - Click “QUIT” button
- Note
 - ISPGUINX is not Windows 7 (64 bit) compatible
 - Install module on Windows 32 bit system and copy to Windows 7
 - Or open IBM PMR, have IBM uncompress modules and email to you



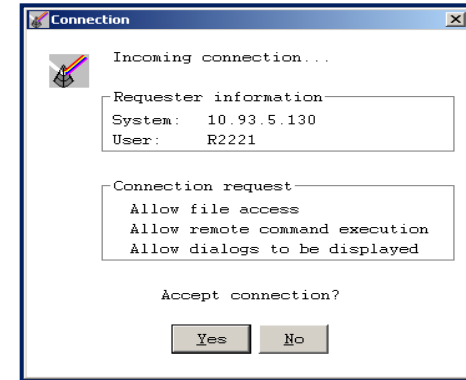
Activation

- Once installed, click on “**WSA.exe**”
- Obtain the workstation IP address
 - From Windows
 - Click “**Start**”
 - Select “**Run**”
 - Insert “**CMD**”
 - Type in “**IPCONFIG**”
 - From ISPF type in “**WSCON**”
 - Type “**1**” for “**Workstation Connection**”
 - Insert “**TCP/IP Address**”

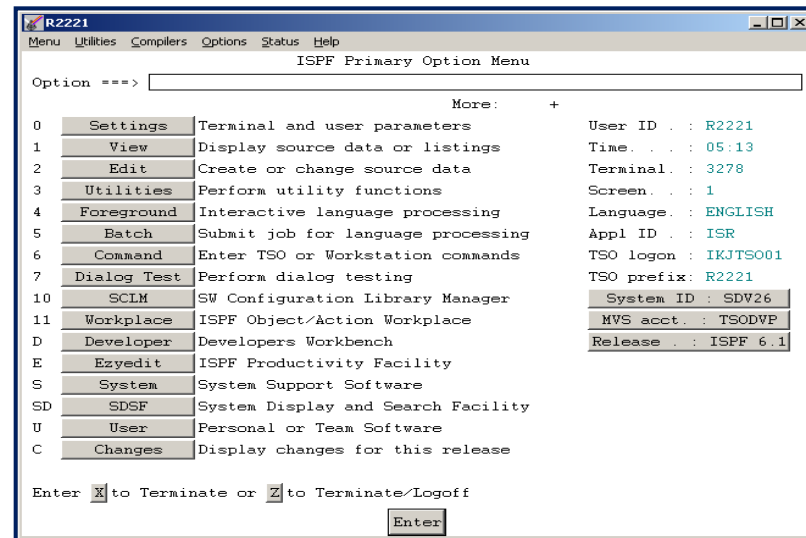


Activation

- Windows “**Connection**” screen will appear
 - Click “**Yes**” button



- ISPF in GUI mode

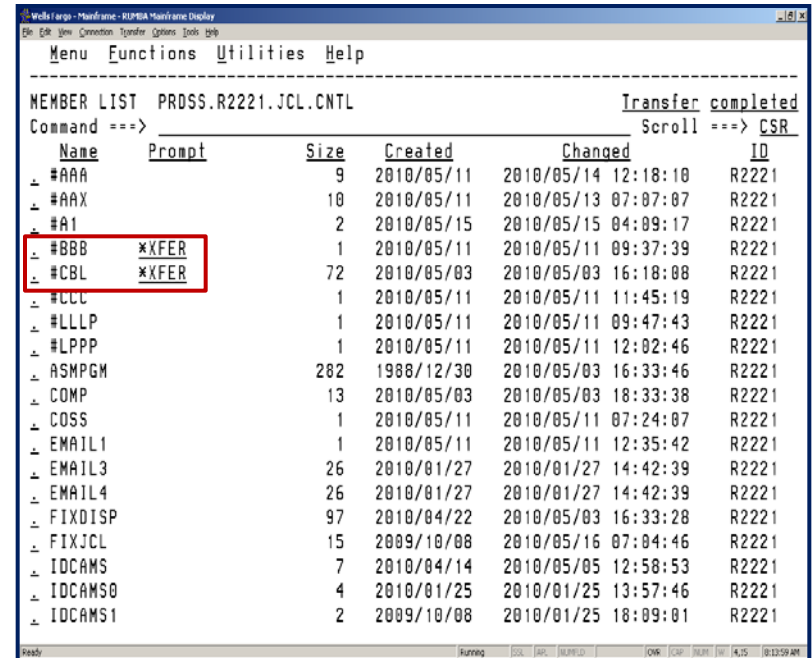
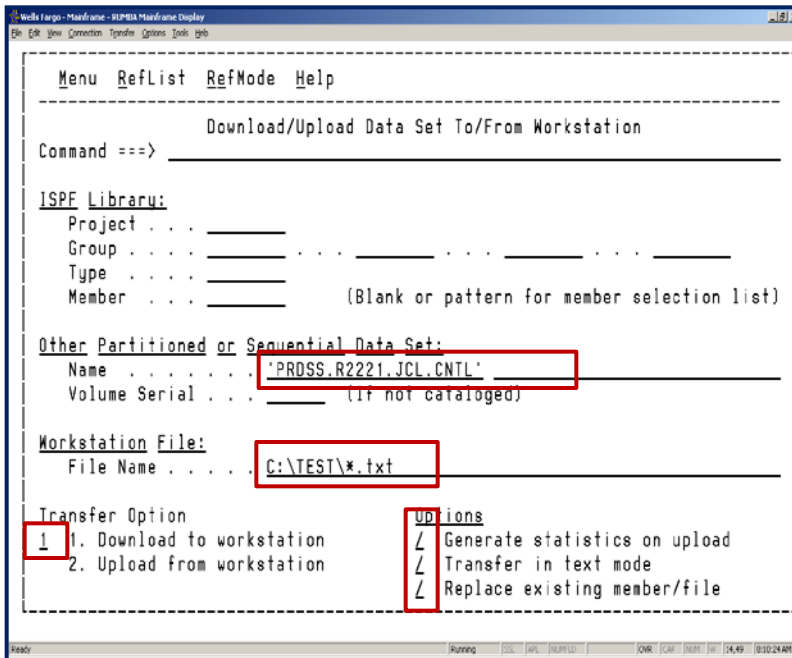


Commands

- **SPLIT** or **START** creates new GUI screen
- **SWAP LIST** display and select screen
- **SWITCH [3270|GUI]** between modes
- **WSDISCON** terminates GUI session

ISPF 3.7.2 – Transfer Files

- ISPF 3.7.2 is an easy way to transfer PDS members to a PC folder



Links

- [ISPF – V1R13 – Dialog Developer’s Guide](#)
- [ISPF – V1R13 – Dialog Tag Language Guide](#)
- [ISPF – V1R13 – Edit and Edit Macros](#)
- [ISPF – V1R13 – Messages and Codes](#)
- [ISPF – V1R13 – Planning and Customization](#)
- [ISPF – V1R13 – Reference Summary](#)
- [ISPF – V1R13 – Services Guide](#)
- [ISPF – V1R13 – Users Guide – Vol 1](#)
- [ISPF – V1R13 – Users Guide – Vol 2](#)

- [REXX – V1R2 – User's Guide](#)
- [REXX – V1R13 – Reference](#)
- [REXX – V1R13 – UNIX System Services](#)

- [SDSF – REXX](#)
- [SDSF – V1R13 – Operations and Customization](#)

Any questions or comments please contact

Bruce Koss

Wells Fargo Mainframe Operating Systems Support

Bruce.Koss@WellsFargo.com

(704) 600-8416