#SHAREorg

SHARE

DFSMS:Intermediate Understanding Someone Else's ACS Routines

Neal Bohling
DFSMS Defect Support, IBM
bohling@us.ibm.com

August 14, 2013
Session# 13158

SHARE in Boston

Agenda

- The Problem
- ACS Review
- Tools and Prep
- Two Scenarios
  - Debugging
  - Adding
- General ACS Tips

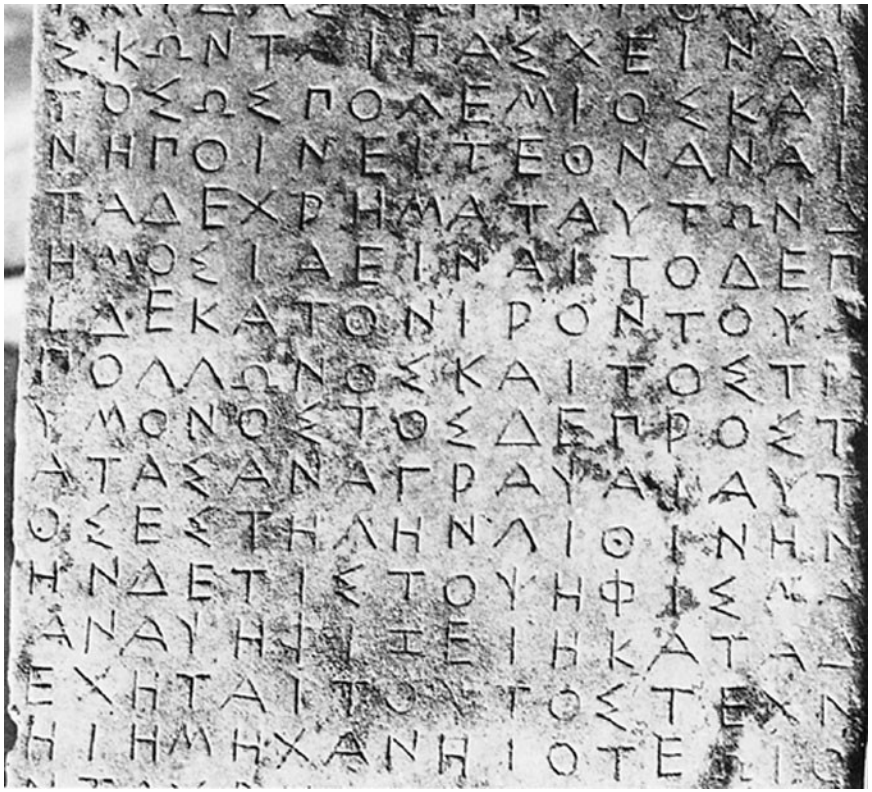
```
IF &LABEL ¬= &DSN(3)
THEN
DO
WRITE '&LABEL NOT SET AS INDICATED BY &DSN(3)'
WRITE '&LABEL NOT VERIFIED BY DATACLAS'
EXIT CODE(16)
END
ELSE
DO
WRITE '&LABEL = '&LABEL' VERIFIED BY DATACLAS'
END
IF &DSN(4) = 'ONE'
THEN
DO
IF &FILENUM ¬= 1
THEN
DO
WRITE '&FILENUM NOT 1 AS INDICATED BY &DSN(4)'
WRITE '&FILENUM NOT VERIFIED BY DATACLAS'
EXIT CODE(16)
END
ELSE
DO
WRITE '&FILENUM NOT 1 AS INDICATED BY &DSN(4)'
WRITE '&FILENUM NOT VERIFIED BY DATACLAS'
EXIT CODE(16)
END
ELSE
DO
WRITE 'DATACLAS ACS ROUTINE -
VERIFIED &FILENUM = '&FILENUM'  '
END
END
ELSE
IF &DSN(4) = 'TWO'
THEN
DO
IF &FILENUM ¬= 2
THEN
DO
WRITE '&FILENUM NOT 2 AS INDICATED BY &DSN(4)'
WRITE '&FILENUM NOT VERIFIED BY DATACLAS'
EXIT CODE(16)
END
ELSE
DO
WRITE '&FILENUM NOT 2 AS INDICATED BY &DSN(4)'
WRITE '&FILENUM NOT VERIFIED BY DATACLAS'
WRITE 'DATACLAS ACS ROUTINE -
VERIFIED &FILENUM = '&FILENUM'  '
END
END
```

The Problem

- Even well-written routines need updated
- Updates are not always as well-written
- Special clauses are added
- Over time, these updates cloud the original purpose

```
IF &LABEL EQ &DSN(3)
THEN
IF &DSN(4) = 'TWO'
THEN
DO
WRITE '&LABEL NOT SET AS INDICATED BY &DSN(3)'
WRITE '&LABEL NOT VERIFIED BY DATACLAS'
IF &FILENUM ¬= 2
WRITE '&LABEL NOT SET AS INDICATED BY &DSN(3)'
EXIT CODE(16)
WRITE '&LABEL NOT VERIFIED BY DATACLAS'
END
ELSE
DO
WRITE '&FILENUM NOT 2 AS INDICATED BY
&DSN(4)'
END
ELSE
DO
WRITE '&LABEL VERIFIED BY DATACLAS'
END
IF &LABEL = 'ONE' &DSN(3) = '&LABEL' VERIFIED BY DATACLAS'
THEN
THEN
IF &DSN(4) = 'ONE'
THEN
DO
IF &FILENUM = 1
WRITE '&LABEL NOT SET AS INDICATED BY &DSN(3)'
THEN
WRITE '&LABEL NOT VERIFIED BY DATACLAS'
DO
IF &FILENUM = 1
EXIT CODE(16)
WRITE '&FILENUM NOT 1 AS INDICATED BY &DSN(4)'
THEN
END
DO
WRITE '&FILENUM NOT VERIFIED BY DATACLAS'
ELSE
WRITE '&FILENUM NOT 1 AS INDICATED BY &DSN(4)'
DO WRITE END &FILENUM NOT VERIFIED BY DATACLAS'
WRITE &LABEL = '&LABEL' VERIFIED BY DATACLAS'
END
DO
WRITE DATACLAS ACS ROUTINE -
IF &DSN(4) = 'ONE'
THEN DO
WRITE END DATACLAS ACS ROUTINE -
DO
VERIFIED &FILENUM = '&FILENUM'
ELSE IF &FILENUM = 1
END &DSN(4) THEN
ELSE
THEN
DO
IF &DSN(4) = 'TWO' WRITE &FILENUM NOT 1 AS INDICATED BY
&DSN(4)
IF &FILENUM = 2
THEN WRITE &FILENUM NOT VERIFIED BY DATACLAS'
DO
IF &FILENUM EXIT CODE(16)
THEN WRITE &FILENUM NOT 2 AS INDICATED BY &DSN(4)'
END
WRITE &FILENUM NOT VERIFIED BY DATACLAS'
DO
ELSE
WRITE &FILENUM NOT 2 AS INDICATED BY &DSN(4)'
WRITE DO &FILENUM NOT VERIFIED BY DATACLAS'
DATACLAS ACS ROUTINE -
VERIFIED &FILENUM = '&FILENUM'
END
```

The Problem



The Solution!

- Unfortunately there is no easy solution
  - But there are things you can do to make it easier!

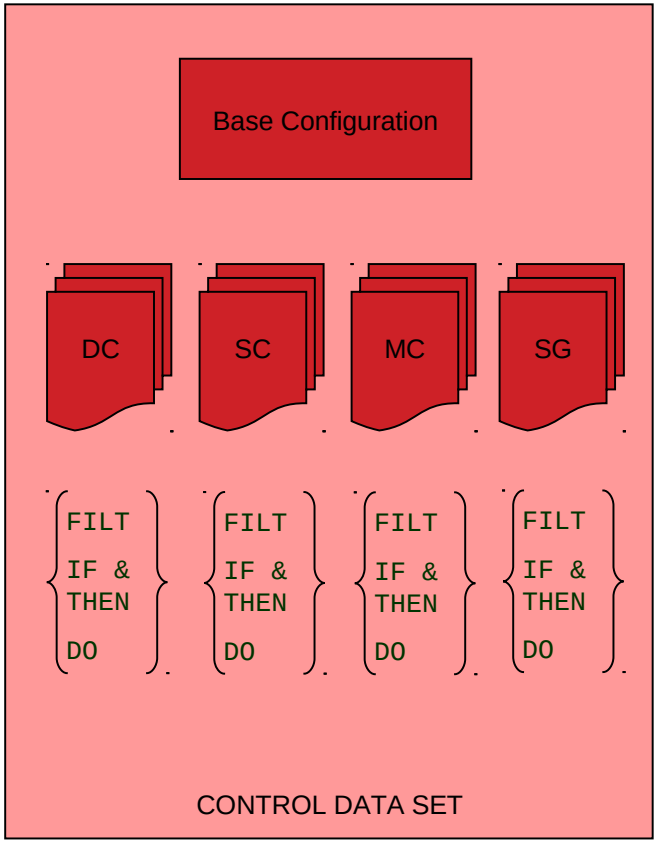
- The **Purpose of this Presentation:**
  - Demonstrate how to find the info you need
  - Tips and tricks for making current ACS easier to read
  - Discuss concepts for good ACS

Part 1 – ACS Review

SMS Review

- SMS has 4 classes:
  - **Data class** – assigns allocation defaults (like size, volcnt)
  - **Storage class*** – assigns performance attributes
  - **Management class** – defines backup characteristics
  - **Storage group*** – groups of volumes
- Each class has it's own ACS routine

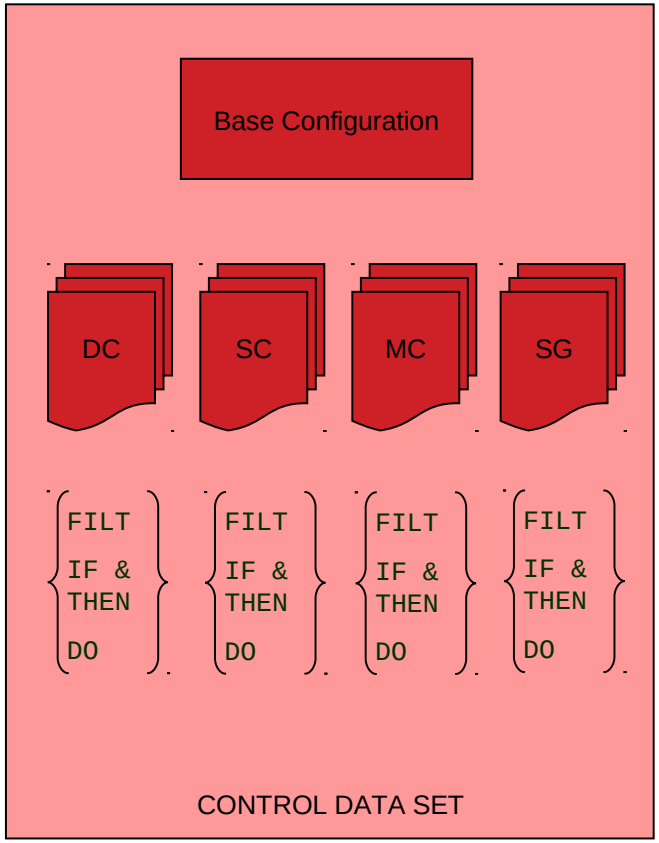
*required to be SMS-managed



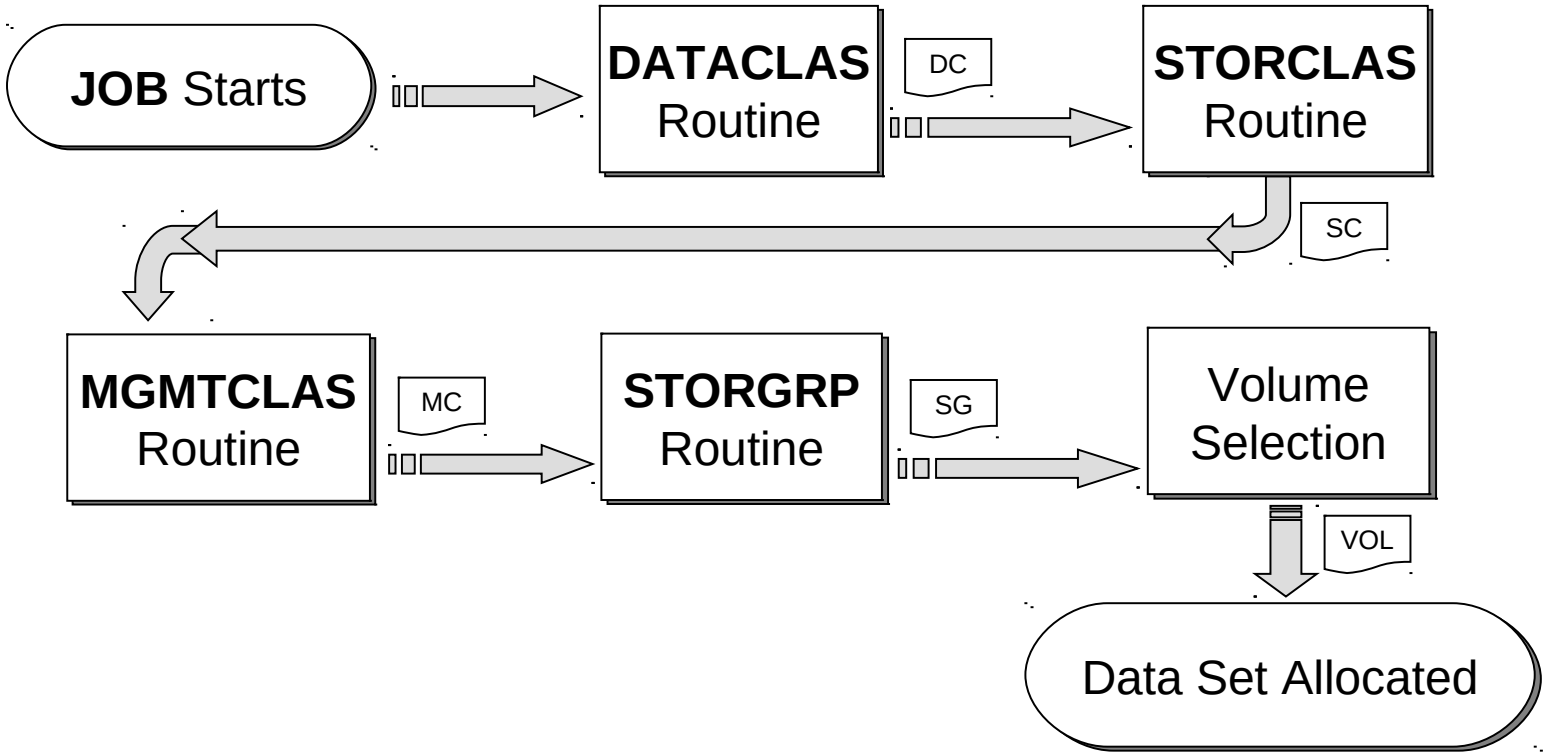
What is ACS?

- Automated Class Selection (ACS)
- User-defined rules
- Assigns SMS classes

- Basically: **YOU** tell **SMS** how to act



Processing Order



ACS Syntax

- Always start with a PROC
- END everything – DO, PROC, SELECT
- IF needs THEN
- SELECT should have WHEN and OTHERWISE
- /* Comments look like this */ - but don't nest them
- Literals are in quotes: 'MY.DATASET' or '3390'
- Masks are not in quotes: MY.* or 33*
- +/- continue literals to the next line

Sample ACS

```
PROC STORCLAS
FILTLIST DBVOLS INCLUDE(IMS*,DB2*)                                /* ALL DATABASE VOLUMES */
                EXCLUDE('IMS053','DB2007')
FILTLIST DBJOBS INCLUDE(IMS*,PROD*,ACCT*)                          /* ALL DATA BASE JOBS */
FILTLIST VALID_UNITS
INCLUDE('3330','3340','3350','3375','3380','3390','SYSDA','')      /* VALID UNITS FOR SMS */
IF &UNIT ^= &VALID_UNITS
   THEN DO
           SET &STORCLAS = ''
           WRITE 'INVALID UNIT TYPE FOR SMS ALLOCATION'
           EXIT
        END
SELECT
       WHEN (&DSN = SYS1.**)                                          /* SYSTEM DATA */
            SET &STORCLAS = 'SYSTEM'
       WHEN ((&ALLVOL = &DBVOLS) && (&JOB = &DBJOBS))                /* DATABASE DATA */
            SET &STORCLAS = 'DBPOOL'
       WHEN ((&DSN(3) = 'CLEAR') | (&ANYVOL ^= TSO*))                /* NON-SMS DATA */
            SET &STORCLAS = ''
       WHEN (&DEF_STORCLAS ^= '')                                    /* IF DEFAULTS EXIST */
            SET &STORCLAS = &DEF_STORCLAS;
       OTHERWISE SET &STORCLAS = 'COMMON'                            /* ALL OTHER DATA */
END
```

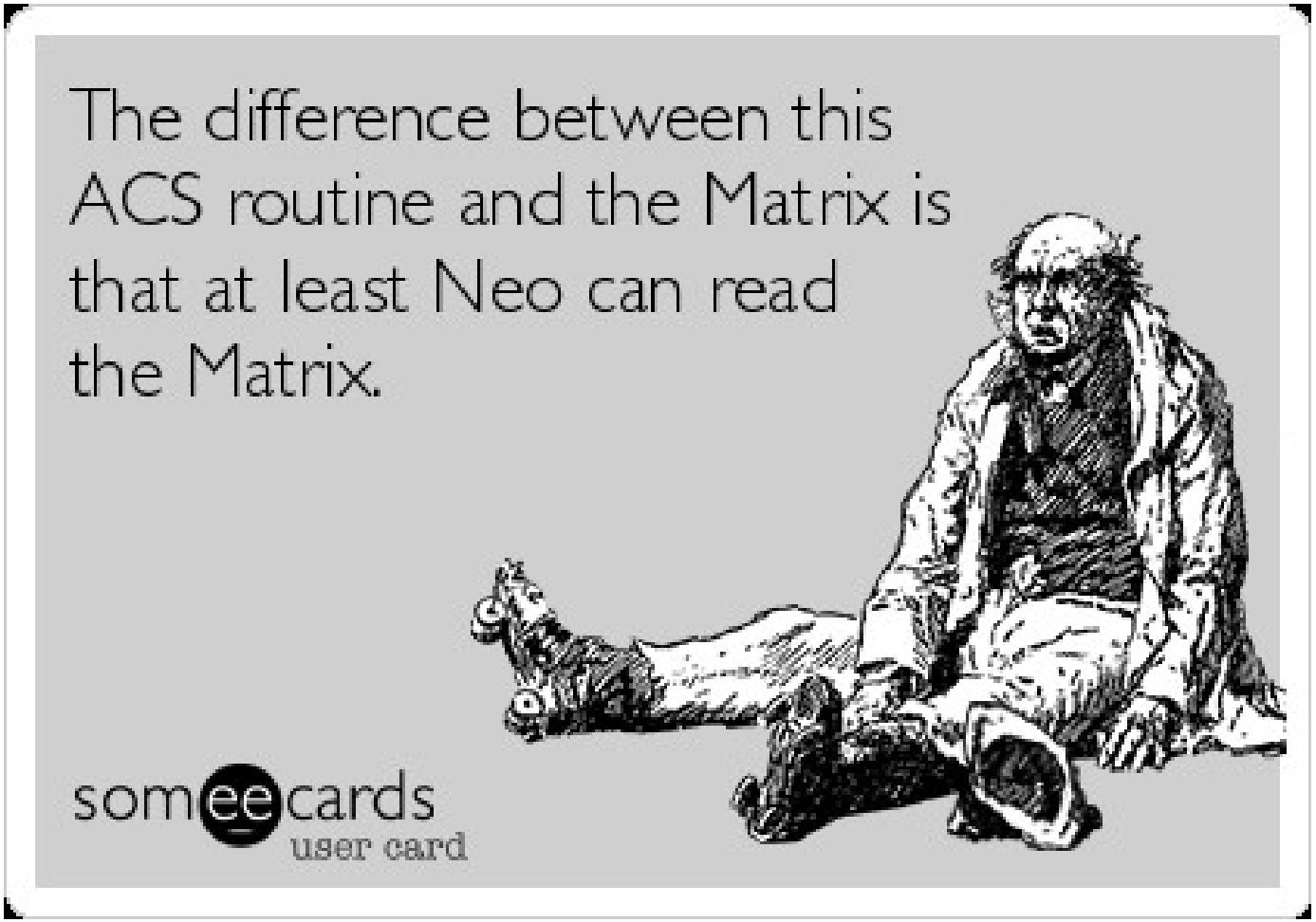
ACS Philosophy

- Several Approaches to ACS routines:

Speedy

Clever

Maintainable



Part 2 – Preparation and Tools

Preparation:

- **Make a copy of your ACS and SCDS**
  - For safety – do any edits on the copy
- Indent
  - Indent after every DO, IF, SELECT
  - Remove indent at every END, ELSE
- Prepare a MAP
  - Helps identify how each class gets set
- Comment where you can -
  - /* document what you already know */
- Add WRITE statements
  - To help trace through routines

Indent Example

Old:

```
IF &DSN(1) = 'TEST' THEN DO
IF &UNIT ^= &VALID_UNITS
  THEN DO
SET &STORCLAS = ''
EXIT
END
ELSE SET &STORCLAS = 'TEST'
END
```

New:

```
/* Check for test data */
IF &DSN(1) = 'TEST' THEN DO
   /* Check if UNITs are VALID */
   IF &UNIT ^= &VALID_UNITS THEN DO
      /* if invalid, set NULL */
      SET &STORCLAS = ''
      WRITE 'INVALID UNIT TYPE'
      EXIT
   END
   ELSE DO
      SET &STORCLAS = 'TEST'
      WRITE 'TEST DATA'
   END
END
```

Making a MAP

- Spreadsheet
- Variables across the top
- Classes down the side

- Allows you to classify how data gets into those classes
- Helps identify groups

	A	B	C	D	E	F	G	H	I
1		&DSTYPE	&JOB	&DSORG	$DSN(2)	&DSTYPE	&SIZE	&USER	&DATACLAS
2	FLATSM								
3	FLATBIG								
4	LIBS								
5	VSAM								
6	CICSVS								
7	TEMPS								
8	JACKS								
9	WRONGDC								
10	ADMIN								
11									

Tools

- **ISMF Option 7.4 – ACS Testing**
  - Define a set of tests
  - Run them and review output
- **NaviQuest (ISMF Option 11)**
  - Very similar, but you can use lists of your actual data sets
  - We'll briefly discuss – for more info, see session 14157
- Many text editors have built-in auto-indent

ACS Testing – ISMF 7.4

```
  Panel  Utilities  Help
                          ACS APPLICATION SELECTION

Select one of the following options:
4  1. Edit            - Edit ACS Routine source code
   2. Translate       - Translate ACS Routines to ACS Object Form
   3. Validate        - Validate ACS Routines Against Storage Constructs
   4. Test            - Define/Alter Test Cases and Test ACS Routines
   5. Display         - Display ACS Object Information
   6. Delete          - Delete an ACS Object from a Source Control Data Set



If Display Option is Chosen, Specify:

  CDS Name  . . 'ACTIVE'
                              (1 to 44 Character Data Set Name or 'Active')



Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

ACS Testing – Define Testcase

```
  Panel  Utilities  Help
                                ACS TEST SELECTION

Select one of the following Options:

  1  1. DEFINE       - Define an ACS Test Case
     2. ALTER        - Alter an ACS Test Case
     3. TEST         - Test ACS Routines


If DEFINE or ALTER Option is Chosen, Specify:

  ACS Test Library . . 'NEAL.SMS.ACS'
  ACS Test Member  . . TESTA___
```

ACS Testing – Define Testcase

```
Panel  Utilities  Scroll  Help
                          ACS TEST CASE DEFINE                    Page 1 of 4

ACS Test Library  : NEAL.SMS.ACS
ACS Test Member . : TESTA

To DEFINE ACS Test Case, Specify:
  Description ==> SC TESTING
  Expected Result SC = SMS
  DSN (DSN/Collection Name) . . NEAL.SMS.MYDATA
  MEMN (Object Name)  . . . . .
  Sysname . . . SYSTEM1   Xmode . . . .  ____      Def_dataclas  . .  ________
  Sysplex . . . ________  ACSenvir  . .  ALLOC     Def_mgmtclas  . .  ________
  DD  . . . . . TESTDD    Dataclas  . .  ________  Def_storclas  . .  ________
  Dsorg . . . . PS        Mgmtclas  . .  ________  Dsntype . . . . .  _____
  Recorg  . . . __        Storclas  . .  ________    If Ext  . . . .  _
  Dstype  . . . ____      Storgrp . . .  ________  Seclabel  . . . .  ________
  Dsowner . . . ________  Size  . . . .  1000      Space_Type  . . .  TRK
  Expdt . . . . _______   Maxsize . . .  __________ Second_Qty  . . .  15
  Retpd . . . . _____     Blksize . . .  __________
Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

ACS Testing – Running a Test

```
  Panel  Utilities  Help
 ------------------------------------------------------------------------------
                              ACS TEST SELECTION            MEMBER TESTA SAVED

Select one of the following Options:

  3  1. DEFINE       - Define an ACS Test Case
     2. ALTER        - Alter an ACS Test Case
     3. TEST         - Test ACS Routines


If DEFINE or ALTER Option is Chosen, Specify:

  ACS Test Library . . 'NEAL.SMS.ACS'
  ACS Test Member  . . TESTA
```

ACS Testing – Running a Test

```
  Panel  Utilities  Help
                                TEST ACS ROUTINES

To Perform ACS Testing, Specify:

  CDS Name . . . . . . 'ACTIVE'
                          (1 to 44 Character Data Set Name or 'Active')
  ACS Test Library . . 'NEAL.SMS.ACS'
  ACS Test Member  . . TESTA    (fully or partially specified or * for all
                          members)
  Listing Data Set . . TEST.LISTING
                          (1 to 44 Character Data Set Name or Blank)

Select which ACS Routines to Test:

  DC ===> Y  (Y/N)  SC ===> Y  (Y/N)  MC ===> Y  (Y/N)  SG ===> Y  (Y/N)



Use ENTER to Perform Verification and Testing;
Use HELP Command for Help; Use END Command to Exit.
```

ACS Testing - Results

```
 BROWSE    USER2.TEST.LISTING
********************************************************************** Top of Data ************>
                                ACS TESTING RESULTS

CDS NAME          : ACTIVE
ACS ROUTINE TYPES: DC SC MC SG
ACS TEST LIBRARY : NEAL.SMS.ACS

  ACS TEST
  MEMBER        EXIT CODE   RESULTS
  ---------     ----------  ------------------------------------
  DESCRIPTION: SC TESTING
  EXPECTED RESULT: SC = SMS
  TESTA                  0  DC = STD
    MSG : DATACLAS=STD
                         0  SC = SMS
    MSG : STORCLAS=SMS
                         0  MC = NULL VALUE ASSIGNED
                         0  SG = SG1


ACS TESTING RC:  00
********************************************************************** Bottom of Data **********>
```

NaviQuest – ISMF 11

- Allows you to build test cases based on real data
  - ISMF Lists, DCOLLECT data, SMF data
- ACS Testing can be run using those test cases
- Can compare tests before / after changes
- Will generate Reports
- **Can be done in batch**

- Covered in session 14157

Computers are like
Old Testament gods;
lots of rules and no mercy.”

- Joseph Campbell

#SHAREorg

SHARE
Technology · Connections · Results

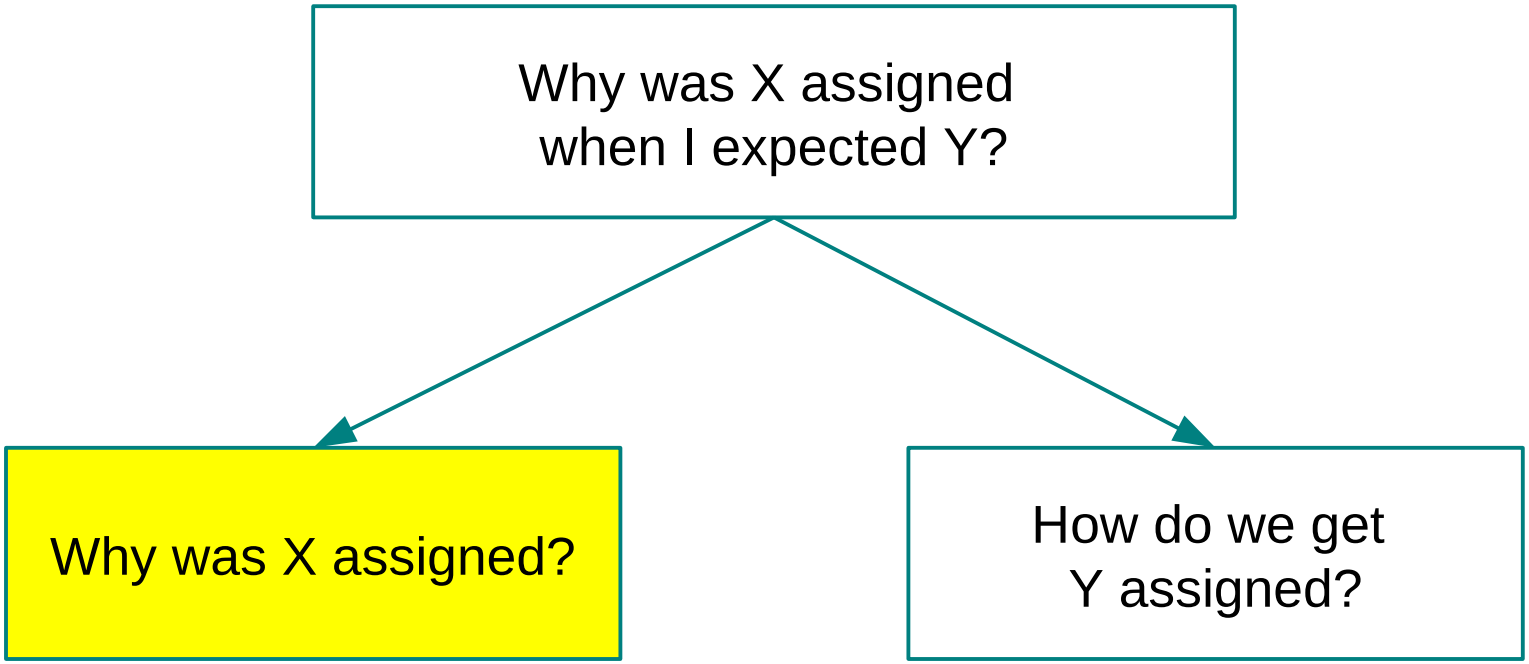
Part 3 – Digging In
"No more procrastinating"

SHARE in Boston

The Common Question

“Why was X assigned when I expected Y?”

Break it Down



Why was X assigned?

- Two ways to solve this:

- **Start at the Beginning**
  - Trace ACS
  - Take each IF that matches
  - End when you find the right SET
- Pros / Cons
  - Can be long
  - There could be multiple paths
  - Straightforward

- **Start at the End**
  - Find all SETs that match
  - Work backwards, noting IF requirements
  - Eliminate redundant requirements
- Pros / Cons
  - Shorter
  - Need less DS information

Working Backwards - Example

- Jack's data set was assigned SC “WrongDC”
- DSNAME : 'SPECIAL.JKTEST.NONVSAM'

- “WRONGDC” this gets set in 2 places
- IF statements just before each:
  - IF &DSN(2) = 'JKTEST'
  - IF &DATACLAS = ''
- IF statements before those:
  - IF &JOB = &CICSJOBS
  - IF &JOB /= &CICSJOBS

```
PROC DATACLAS

FILTLIST ADMINS   INCLUDE('BOB','LARRY','MOE')
FILTLIST CICSJOBS INCLUDE(CICS*,'CISPECL')

/* DATA CLASSES */
/* FLATSM  - flat files, small, < 50mb */
/* FLATBIG - flat files, big > 50mb    */
/* LIBS    - PDS, PDSE                 */
/* VSAM    - VSAM files                */
/* CICSVS  - VSAM files for CICS       */
/* TEMPS   - temporary                 */
/* JACKS   - jacks testing DC          */
/* WRONGDC - invalid combination       */

IF &DSTYPE = 'TEMP' THEN SET &DATACLAS = 'TEMPS'

IF &JOB = &CICSJOBS THEN DO
             IF &DSORG = 'VS' THEN SET &DATACLAS = 'CICSVS'
             IF &DSN(2) = 'JKTEST' THEN SET &DATACLAS = 'JACKS'
             ELSE SET &DATACLAS = 'WRONGDC'
ELSE DO
             IF &DSORG = 'PS' THEN DO
                          IF &DSTYPE = 'TEMP' THEN SET &DATACLAS = 'TEMP'
                          IF &SIZE > 50MB THEN SET &DATACLAS = 'FLATSM'
                          ELSE SET &DATACLAS = 'FLATBIG'
             END

             IF &DSORG = 'PO' THEN SET &DATACLAS = 'LIBS'

             IF &USER = &ADMINS THEN DO
                          SET &DATACLAS = 'ADMIN'
                          EXIT
             END

             IF &DSORG = 'VS' THEN DO
                          SET &DATACLAS = 'VSAM'
                          IF &DSTYPE = 'TEMP' THEN SET &DATACLAS = 'TEMP'
                          END

             IF &JOB = 'JKTEST' THEN SET &DATACLAS = 'JACKS'
             IF &DATACLAS = '' THEN SET &DATACLAS = 'WRONGDC'
END
END
```

```
IF &DSTYPE = 'TEMP' THEN SET &DATACLAS = 'TEMPS'

IF &JOB = &CICSJOBS THEN DO
        IF &DSORG = 'VS' THEN SET &DATACLAS = 'CICSVS'
        IF &DSN(2) = 'JKTEST' THEN SET &DATACLAS = 'JACKS'
        ELSE SET &DATACLAS = 'WRONGDC'
ELSE DO
        IF &DSORG = 'PS' THEN DO
                IF &DSTYPE = 'TEMP' THEN SET &DATACLAS = 'TEMP'
                IF &SIZE > 50MB THEN SET &DATACLAS = 'FLATSM'
                ELSE SET &DATACLAS = 'FLATBIG'
        END

        IF &DSORG = 'PO' THEN SET &DATACLAS = 'LIBS'

        IF &USER = &ADMINS THEN DO
                SET &DATACLAS = 'ADMIN'
                EXIT
        END

        IF &DSORG = 'VS' THEN DO
                SET &DATACLAS = 'VSAM'
                IF &DSTYPE = 'TEMP' THEN SET &DATACLAS = 'TEMP'
                END

        IF &JOB = 'JKTEST' THEN SET &DATACLAS = 'JACKS'
        IF &DATACLAS = '' THEN SET &DATACLAS = 'WRONGDC'
END
END
```

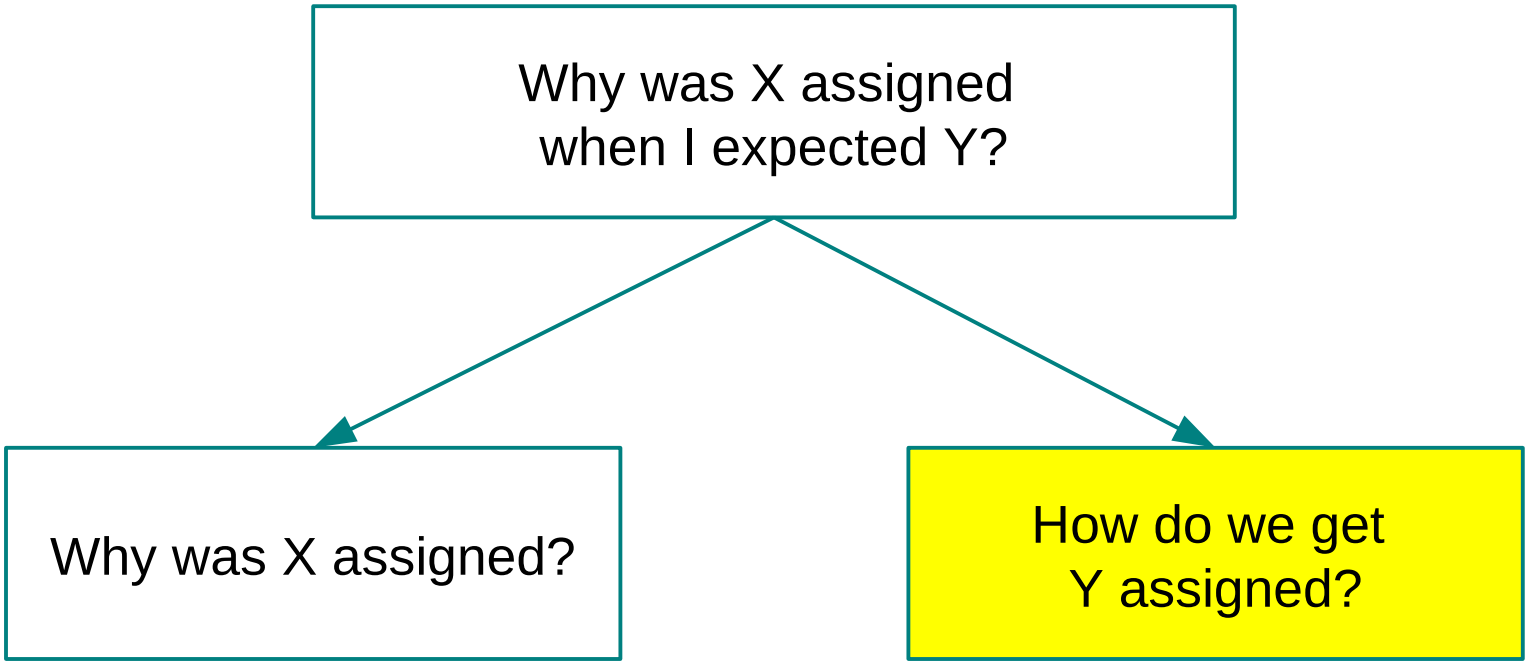
Working Backwards - Example

- Write the full rules:
  - IF &JOB = &CICSJOB AND &DSN(2) /= 'JKTEST'
  - IF &JOB /= &CICSJOB AND &DATACLAS = '' (not set)

- Now we have the 2 cases where WRONGDC gets set
- Update the MAP

	A	B	C	D	E	F	G	H	I
1		&DSTYPE	&JOB	&DSORG	$DSN(2)	&DSTYPE	&SIZE	&USER	&DATACLAS
2	WRONGDC		&CICSJOBS		NOT JKTEST				
3	WRONGDC		NOT &CICSJOBS						null
4									

Break it Down



How do we get Y assigned?

- Let's say the DS is supposed to have DC = 'JACKS'
- Identify all places JACKS is set:
  - IF &DSN(2) = 'JKTEST'
  - IF &JOB = 'JKTEST'
- Second-level tests:
  - IF &JOB = &CICSJOBS
  - IF &JOB /= &CICSJOBS

```
IF &DSTYPE = 'TEMP' THEN SET &DATACLAS = 'TEMPS'

IF &JOB = &CICSJOBS THEN DO
        IF &DSORG = 'VS' THEN SET &DATACLAS = 'CICSVS'
        IF &DSN(2) = 'JKTEST' THEN SET &DATACLAS = 'JACKS'
        ELSE SET &DATACLAS = 'WRONGDC'
ELSE DO
        IF &DSORG = 'PS' THEN DO
                IF &DSTYPE = 'TEMP' THEN SET &DATACLAS = 'TEMP'
                IF &SIZE > 50MB THEN SET &DATACLAS = 'FLATSM'
                ELSE SET &DATACLAS = 'FLATBIG'
        END

        IF &DSORG = 'PO' THEN SET &DATACLAS = 'LIBS'

        IF &USER = &ADMINS THEN DO
                SET &DATACLAS = 'ADMIN'
                EXIT
        END

        IF &DSORG = 'VS' THEN DO
                SET &DATACLAS = 'VSAM'
                IF &DSTYPE = 'TEMP' THEN SET &DATACLAS = 'TEMP'
                END

        IF &JOB = 'JKTEST' THEN SET &DATACLAS = 'JACKS'
        IF &DATACLAS = '' THEN SET &DATACLAS = 'WRONGDC'
END
END
```

How to assign Y - Special note:

- There is an EXIT for combination &USER = &ADMIN
- This creates an IMPLIED condition for getting JACKS:
  - &USER /= &ADMIN
- Now we know how to get what we want:
  - &JOB = &CICSJOB AND &DSN(2) = 'JKTEST'
  - &JOB /= &CICSJOB AND &USER /= &ADMIN AND &JOB = 'JKTEST'

	A	B	C	D	E	F	G	H	I
1		&DSTYPE	&JOB	&DSORG	$DSN(2)	&DSTYPE	&SIZE	&USER	&DATACLAS
2	WRONGDC		&CICSJOBS		NOT JKTEST				
3	WRONGDC		NOT &CICSJOBS						null
4	JACKS		&CICSJOBS		JKTEST				
5	JACKS		JKTEST					NOT &ADMIN	

Getting it Right

- So what do we need to change?
- We're still missing some information:

- Get the data set information for the variables:
  - &JOB = 'MYTEST'
  - &DSN(2) = 'JKTEST'
  - &USER = 'JACK'
  - &DATACLAS = TBD
  - &DSORG = 'DA' (BDAM)

So what went wrong?

- Consolidate:
  - IF &JOB = &CICSJOB
    - IF &DSN(2) = 'JKTEST' – what we want
    - ELSE – what we don't want
  - IF &JOB /= &CICSJOB
    - IF &JOB = 'JKTEST' – what we want
    - IF &DATACLAS = '' - what we don't want
    -
- We know &JOB was NOT &CICSJOB, remove those rules

	A	B	C	D	E	F	G	H	I
1	CLASS	&DSTYPE	&JOB	&DSORG	$DSN(2)	&DSTYPE	&SIZE	&USER	&DATACL
2	WRONGDC		NOT &CICSJOBS						null
3	JACKS		JKTEST					NOT &ADMIN	
4	~~WRONGDC~~		~~&CICSJOBS~~		~~JKTEST~~				
5	~~JACKS~~		~~&CICSJOBS~~		~~JKTEST~~				

What went wrong?

- We now have only 2 tests:
  - IF &JOB = 'JKTEST'
  - IF &DATACLAS = ''

- Expand the second rule by looking at the rules that would let that be true:
  - IF &DSORG /= PS | PO | VS AND &JOB /= 'JKTEST'

What went wrong?

- Adding the Data Set to the Map (some columns hidden)

	A	C	D	E	H
1	CLASS	&JOB	&DSORG	$DSN(2)	&USER
2	WRONGDC	NOT &CICSJOBS and NOT JKTEST	NOT PO|PS|VS		
3	JACKS	JKTEST			NOT &ADMIN
4	DataSet	MYTEST	DA	JKTEST	JACK
5					

Matches WRONGDC

Not Compared

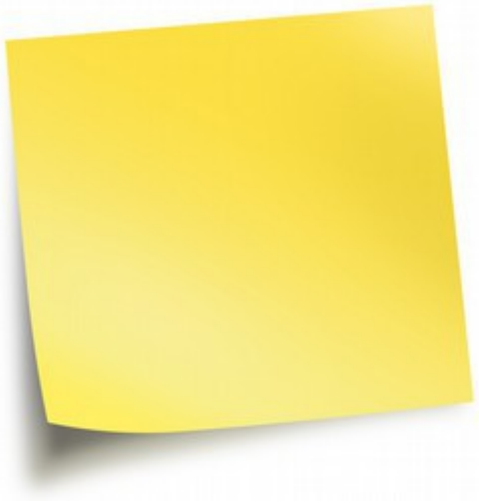
What went wrong?

- If it's not clear yet, here's what happened:
  - Jack was doing his testing the same as for his CICS jobs
  - This job is not CICS, so the rules changed and he didn't realize it
  - To get DC = 'JACKS', for NON-CICS, the JOBNAME needs to be 'JKTEST'

- This was just a simple example – working through the logic can help a lot with bigger, more complex ACS

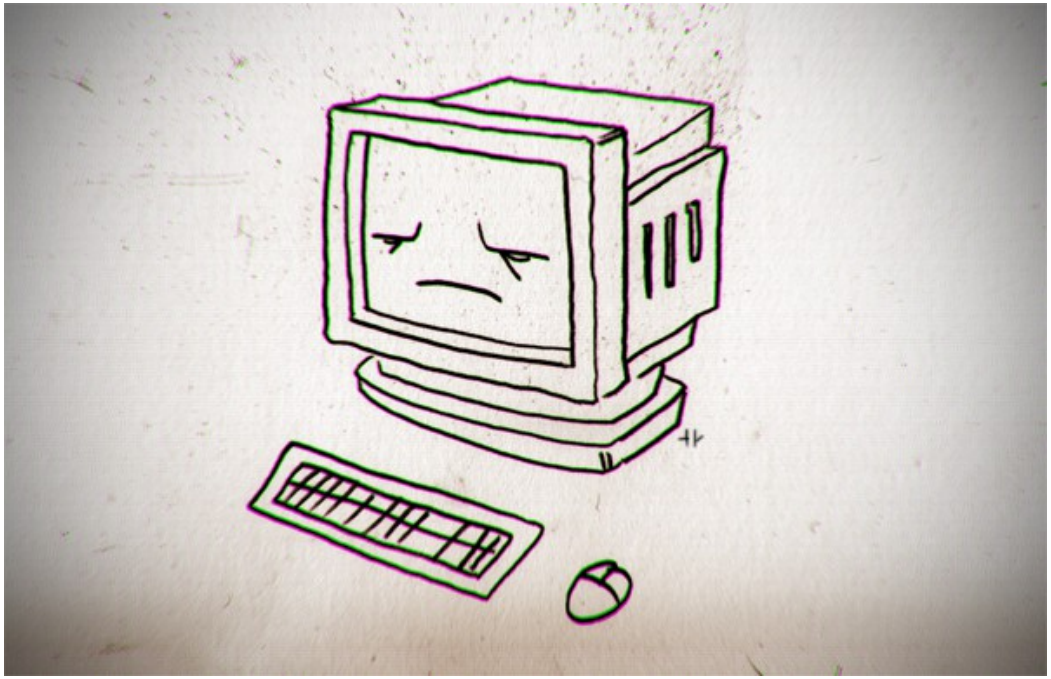
A few notes:

- Implied rules complicate things
  - Using R/W variables adds implied rules
  - Therefore, try to avoid it in your ACS
- EXIT after each set would add implied rules
  - But they would also avoid multiple SETs
- Mutually exclusive rules solves both requirements above
- You could also add strategic WRITES to demonstrate decision logic
- Use ACS testing to run your modified ACS to track logic



Summary:

- Why was X assigned?
- Tracked it backwards from the SET
- Created a MAP to detail conditions
- Compared actual rules to expected result
- Found culprit



“To err is human - and to blame it on a computer is even more so.”

- Robert Orben

Scenario 2

“We added a class, and now we need to update the routines.”

Updating Routines - Overview

- Varying difficulty, depending on new rules
- Simpler with a MAP

- Basic steps:
  - Copy and work from copy!
  - Find logic section that matches
  - Insert new rule
  - Test, fix, test, fix

Updating Routines – Don't

- Don't just add a simple rule to the beginning:
  - IF &DSORG=”VS” THEN SET &DATACLAS = “VSAM”
  - Nothing else will get run
- Don't just add a simple rule to the end:
  - It might never get run
- Don't forget to comment
- Don't use variables that are not used anywhere else
  - You'll end up with non-exclusive paths

Understand the Logic

- How do the routines select classes?
- What variables are used?
- Back to the map:
  - Hopefully each class has only one combination

	A	B	C	D	E	F	G	H	I
1	CLASS	&DSTYPE	&JOB	&DSORG	$DSN(2)	&DSTYPE	&SIZE	&USER	&DATACL
2	WRONGDC		NOT &CICSJOBS and NOT JKTEST	DA				NOT &ADMIN	
3	FLATSM		NOT &CICSJOBS	PS			< 50M		
4	FLATBIG		NOT &CICSJOBS	PS			>= 50M		
5	LIBS		NOT &CICSJOBS	PO					
6	VSAM		NOT &CICSJOBS	VS					
7	TEMPS	TEMP	NOT &CICSJOBS	PS | VS					
8	ADMIN		NOT &CICSJOBS					&ADMIN	
9	JACKS		JKTEST					NOT &ADMIN	
10	CICSVS		&CICSJOB	VS					
11	JACKS		&CICSJOB		JKTEST				
12	WRONGDC		&CICSJOB						
13									

Adding Rules - Example

- New class BDAMSTUF
  - Non-CICS
  - DSORG='DA'
- Add to the MAP:

	A	B	C	D	E	F	G	H	I
1	CLASS	&DSTYPE	&JOB	&DSORG	$DSN(2)	&DSTYPE	&SIZE	&USER	&DATACL
2	WRONGDC		NOT &CICSJOBS and NOT JKTEST	DA				NOT &ADMIN	
3	FLATSM		NOT &CICSJOBS	PS			< 50M		
4	FLATBIG		NOT &CICSJOBS	PS			>= 50M		
5	LIBS		NOT &CICSJOBS	PO					
6	VSAM		NOT &CICSJOBS	VS					
7	TEMPS	TEMP	NOT &CICSJOBS	PS | VS					
8	ADMIN		NOT &CICSJOBS					&ADMIN	
9	JACKS		JKTEST					NOT &ADMIN	
10	CICSVS		&CICSJOB	VS					
11	JACKS		&CICSJOB		JKTEST				
12	WRONGDC		&CICSJOB						
13	BDAMSTUF		NOT &CICSJOBS	DA					
14									

Adding Rules - Example

- Compare to other rules by Variable
  - Only two in play: **&JOB and &DSORG**
- Fits in our NOT &CICSJOB section
- Fits next to other &DSORG tests

	A	B	C	D	E	F	G	H	I
1	**CLASS**	**&DSTYPE**	**&JOB**	**&DSORG**	**$DSN(2)**	**&DSTYPE**	**&SIZE**	**&USER**	**&DATACLAS**
2	**CICSVS**		&CICSJOB	VS					
3	**JACKS**		&CICSJOB		JKTEST				
4	**WRONGDC**		&CICSJOB						
5	**JACKS**		JKTEST					NOT &ADMIN	
6	**BDAMSTUF**		NOT &CICSJOBS	DA					
7	**LIBS**		NOT &CICSJOBS	PO					
8	**FLATSM**		NOT &CICSJOBS	PS			< 50M		
9	**FLATBIG**		NOT &CICSJOBS	PS			>= 50M		
10	**TEMPS**	TEMP	NOT &CICSJOBS	PS | VS					
11	**VSAM**		NOT &CICSJOBS	VS					
12	**ADMIN**		NOT &CICSJOBS					&ADMIN	
13	**WRONGDC**		NOT &CICSJOBS and NOT JKTEST	DA				NOT &ADMIN	

```
IF &DSTYPE = 'TEMP' THEN SET &DATACLAS = 'TEMPS'

IF &JOB = &CICSJOBS THEN DO
        IF &DSORG = 'VS' THEN SET &DATACLAS = 'CICSVS'
        IF &DSN(2) = 'JKTEST' THEN SET &DATACLAS = 'JACKS'
        ELSE SET &DATACLAS = 'WRONGDC'
ELSE DO
        IF &DSORG = 'PS' THEN DO
                IF &DSTYPE = 'TEMP' THEN SET &DATACLAS = 'TEMP'
                IF &SIZE > 50MB THEN SET &DATACLAS = 'FLATSM'
                ELSE SET &DATACLAS = 'FLATBIG'
        END

        IF &DSORG = 'PO' THEN SET &DATACLAS = 'LIBS'

        IF &USER = &ADMINS THEN DO
                SET &DATACLAS = 'ADMIN'
                EXIT
        END

        IF &DSORG = 'VS' THEN DO
                SET &DATACLAS = 'VSAM'
                IF &DSTYPE = 'TEMP' THEN SET &DATACLAS = 'TEMP'
                END

        IF &JOB = 'JKTEST' THEN SET &DATACLAS = 'JACKS'
        IF &DATACLAS = '' THEN SET &DATACLAS = 'WRONGDC'
END
END
```

NOT &CICSJOB

&DSORG Section

Adding Rules - Example

- Insert the new rule:
  - IF &DSORG='DA' THEN SET &DATACLAS = 'BDAMSTUF'
- This many IF statements should be a SELECT
- Use the best practices already in use in the routine
  - Don't change practices unless you're ready to rewrite
- Note your update in the CHANGELOG
  - If one doesn't exit, CREATE ONE!

```
PROC DATACLAS

FILTLIST ADMINS   INCLUDE('BOB','LARRY','MOE')
FILTLIST CICSJOBS INCLUDE(CICS*,'CISPECL')

/* DATA CLASSES */
/* FLATSM  - flat files, small, < 50mb */
/* FLATBIG - flat files, big > 50mb    */
/* LIBS    - PDS, PDSE                 */
/* VSAM    - VSAM files                */
/* CICSVS  - VSAM files for CICS       */
/* TEMPS   - temporary                 */
/* JACKS   - jacks testing DC          */
/* WRONGDC - invalid combination       */

/* CHANGE LOG                                        */
/* 2013.08.13 - Added BDAMSTUF to NON-CICS group     */
/*            - Also switched from IF to SELECT block */
```

```
IF &DSTYPE = 'TEMP' THEN SET &DATACLAS = 'TEMPS'

IF &JOB = &CICSJOBS THEN DO
         IF &DSORG = 'VS' THEN SET &DATACLAS = 'CICSVS'
         IF &DSN(2) = 'JKTEST' THEN SET &DATACLAS = 'JACKS'
         ELSE SET &DATACLAS = 'WRONGDC'
ELSE DO
         SELECT ( &DSORG )
                  WHEN ('PS') THEN DO
                           IF &DSTYPE = 'TEMP' THEN SET &DATACLAS = 'TEMP'
                           IF &SIZE > 50MB THEN SET &DATACLAS = 'FLATSM'
                           ELSE SET &DATACLAS = 'FLATBIG'
                  END
                  WHEN ('PO') THEN SET &DATACLAS = 'LIBS'
                  WHEN ('VS') THEN DO
                           SET &DATACLAS = 'VSAM'
                           IF &DSTYPE = 'TEMP' THEN SET &DATACLAS = 'TEMP'
                  END
                  WHEN ('DA') THEN SET &DATACLAS = 'BDAMSTUF'
                  OTHERWISE SET &DATACLAS = 'WRONGDC'
         END /* END SELECT &DSORG */

         IF &USER = &ADMINS THEN DO
                  SET &DATACLAS = 'ADMIN'
                  EXIT
         END
         IF &JOB = 'JKTEST' THEN SET &DATACLAS = 'JACKS'
         IF &DATACLAS = '' THEN SET &DATACLAS = 'WRONGDC'
END
END
```

Test, Fix, Test

- Use the ACS routine tester
- Run a series of tests to verify new changes work
- Run a series of tests to ensure old rules still work
- If not, figure out why and fix
- Repeat

- You can use NaviQuest to build suites of tests
  - See session 14157 – NaviQuest: Streamlining SMS

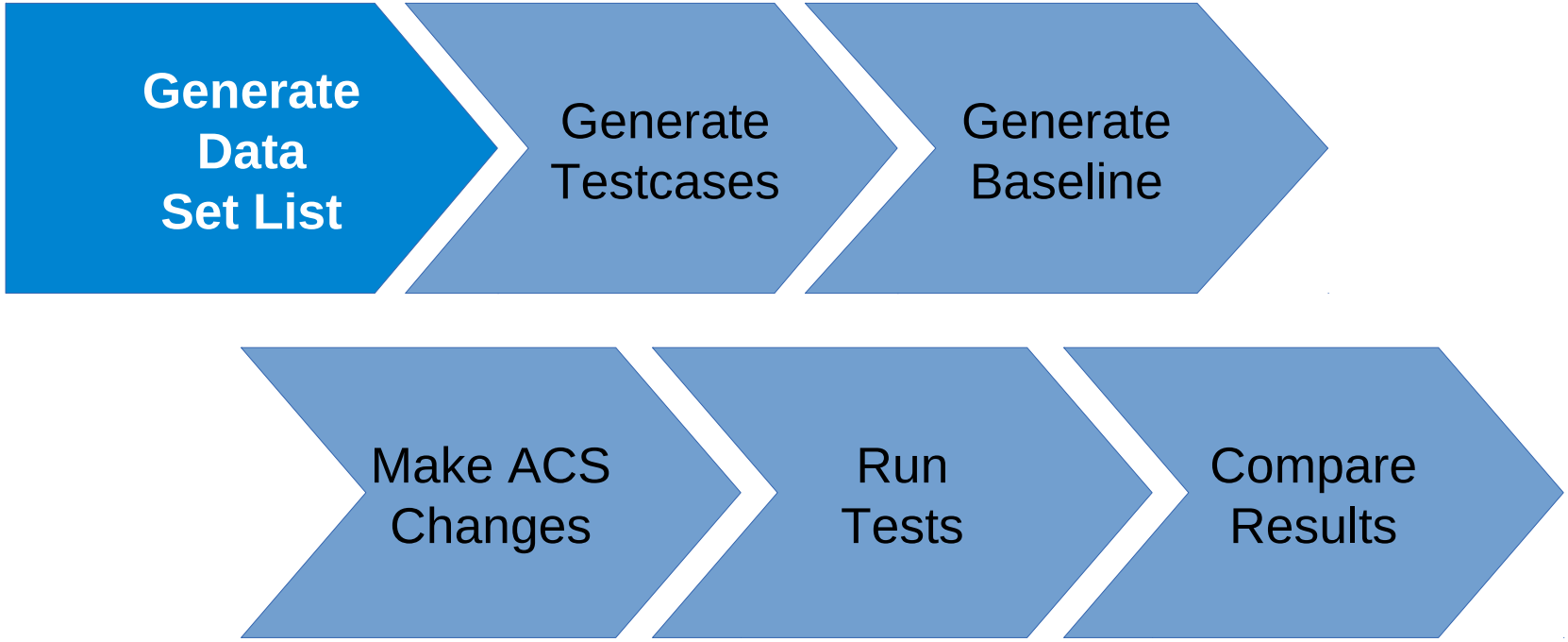
Adding Rules - Summary

- Make a copy
- Understand the logic
- Find relevant section
- Update CHANGELOG
- Add rule
- Avoid breaking anything
- Test, test, test

Summary of Hints and Tips

- Make code more readable by
  - Add WRITE statements to existing ACS
  - Add comments to existing ACS
  - Indent appropriately
  - Renaming FILTLISTS
- If you are going to rewrite
  - Clarify rules BEFORE writing
  - Group rules using nested SELECT / IFs
  - Think about how you might need to expand
  - Comment everything
  - One SET for each class
  - Exit after SET
  - Put most commonly executed paths first
  - Try to make your IF/SELECT statements mutually exclusive
- Remember: the goal is MAINTAINABLE, not clever

ACS Testing - Process



ACS Testing – Generating a Data Set List

- Generate a data set list and save it
  - **ISMF – panel 1 or 2**
  - DCOLLECT – faster than ISMF, good for batch
  - SMF – ACS exit, good for temporary data
  - VMA – good for tape data
- Use NaviQuest to generate test cases
- Generate 'Baseline'

ACS Testing - Generate Data Set List

```
                    ISMF PRIMARY OPTION MENU - z/OS DFSMS V1 R13

 0  ISMF Profile                 - Specify ISMF User Profile
 1  Data Set                     - Perform Functions Against Data Sets
 2  Volume                       - Perform Functions Against Volumes
 3  Management Class             - Specify Data Set Backup and Migration Criteria
```

```
  Panel  Defaults  Utilities  Scroll  Help
                          DATA SET SELECTION ENTRY PANEL              Page 1 of 5

 For a Data Set List, Select Source of Generated List  . . 2  (1 or 2)

   1  Generate from a Saved List          Query Name To
        List Name  . . ________           Save or Retrieve ________

   2  Generate a new list from criteria below
        Data Set Name . . . 'NB.**'
        Enter "/" to select option    _   Generate Exclusive list
        Specify Source of the new list  . . 2     (1 - VTOC, 2 - Catalog)
        1 Generate list from VTOC
           Volume Serial Number . . . ______      (fully or partially specified)
           Storage Group Name . . . . ________    (fully specified)
        2 Generate list from Catalog
           Catalog Name . . . ____________________________________________
           Volume Serial Number . . . ______      (fully or partially specified)
           Acquire Data from Volume . . . . . . . Y  (Y or N)
           Acquire Data if DFSMShsm Migrated  . . Y  (Y or N)
 Use ENTER to Perform Selection; Use DOWN Command to View next Selection Panel;
 Use HELP Command for Help; Use END Command to Exit.
```

ACS Testing – Generate DS List

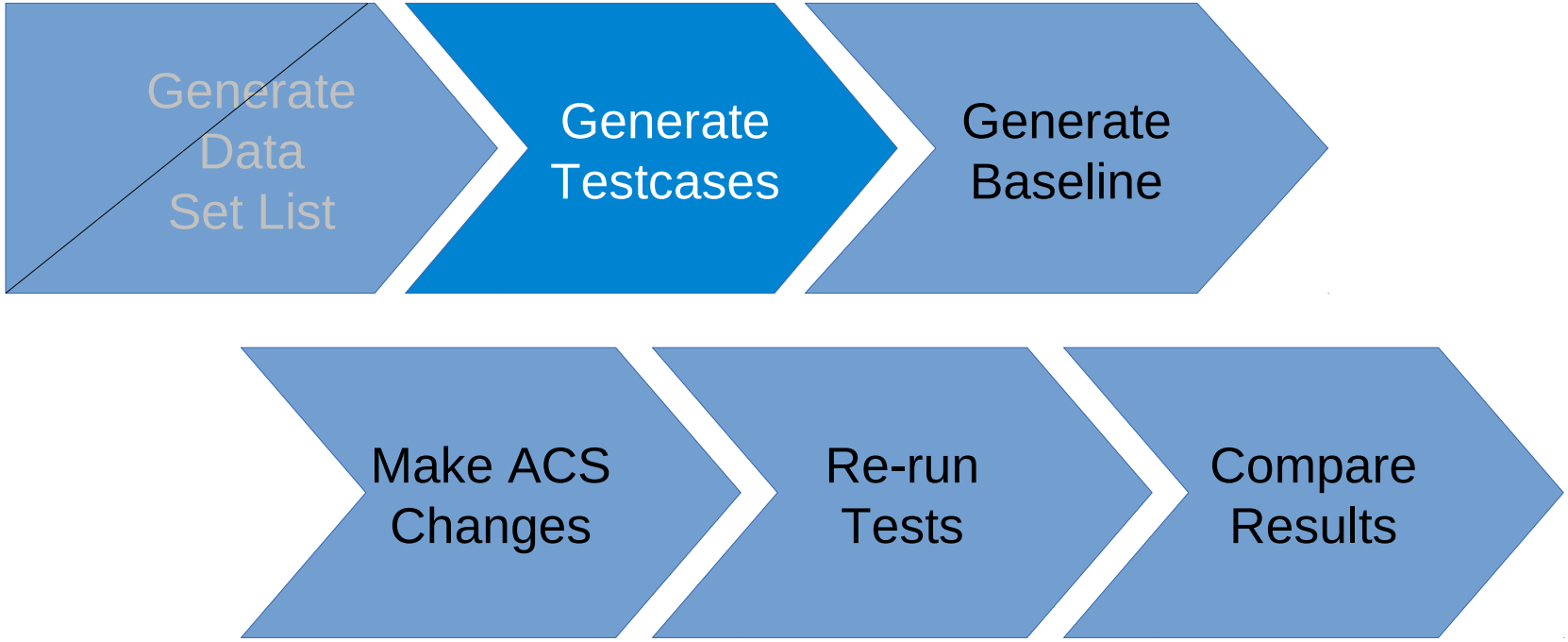
```
  Panel  List  Dataset  Utilities  Scroll  Help
-------------------------------------------------------------------------------
                                                          DATA SET LIST
                                                       Entries 1-8 of 8
Enter Line Operators below:                            Data Columns 4-6 of 42

     LINE                                    ALLOC     % NOT  COMPRESSED
     OPERATOR          DATA SET NAME         USED      USED   FORMAT
    ---(1)----  ------------(2)------------ ---(4)---  -(5)-  ---(6)----
                NB.RLS.TEST1                ---------    ---  ---
                NB.RLS.TEST1.DATA                810K      0  ---
                NB.RLS.TEST1.INDEX                 1K     96  ---
                NB.SMS.TESTA                      55K     93  NO
                NB.SMS.TESTB                      55K     93  NO
                NB.SMS.TESTOUT                    55K     99  NO
                NB.SMSN.TEST1                     55K     93  ---
                NB.SMSN.TEST2                     55K     93  ---
    ----------  ------  -----------  BOTTOM OF  DATA  -----------  ------  ----


Command ===> SAVE NBFILES _                                   Scroll ===> HALF
```

* Note: you can see and edit your saved lists from ISMF Option L (LISTS)

ACS Testing - Process



ACS Testing – Generate Testcases

- ISMF 11.1.1 – NaviQuest generate

```
 9  Aggregate Group          - Specify Data Set Recovery Parameters
10  Library Management       - Specify Library and Drive Configurations
11  Enhanced ACS Management  - Perform Enhanced Test/Configuration Management
C   Data Collection          - Process Data Collection Function
G   Report Generation        - Create Storage Management Reports
```

```
              ENHANCED ACS MANAGEMENT - NaviQuest PRIMARY OPTION MENU

Select one of the following options and press Enter:

  1  Test Case Generation
  2  ACS Test Listings Comparison
  3  Enhanced ACS Test Listing
  4  Test Case Update with Test Results
  5  SMS Report Generation
  6  Model Commands Generation
  7  Batch Testing/Configuration Management
  X  Exit
```

```
                        TEST CASE GENERATION SELECTION MENU

Select the input data to be used and press Enter:

  1  Saved ISMF List
  2  DCOLLECT Data
  3  SMF Data
  4  VMA Extract Data
```

ACS Testing – Generate Testcases

```
  Panel  Help
 ------------------------------------------------------------------------------
              TEST CASE GENERATION FROM SAVED ISMF LIST              Top of data

 To generate test cases, specify the following information and press Enter:
   Saved ISMF List . . . . . . NBFILES   (Data set list)
   Member Name Prefix  . . . . NB        (1 to 4 alpha characters)
   Test Case PDS . . . . . . . 'NEAL.SMS.ACSTST'
     Replace Existing Prefix   N         (Y or N)
 ACS Test Case Variables:
                                                                  More:     +
   Applic  . . . . CICSJOB_          DD  . . . . . . . ________
   Def_dataclas    ________          Def_mgmtclas      ________
   Def_storclas    ________          Filenum . . . . ____
   Group . . . . . ________          Job . . . . . . ________
   Label . . . . . ___               Libname . . . . ________
   Pgm . . . . . . ________          Storgrp . . . . ________
   Retpd . . . . . _____             Sysplex . . . . ________

 Use DOWN Command to Scroll Forward; Use UP Command to Scroll Backward;
 Use HELP Command for Help; Use END Command to Exit.
```

* there is a special prefix – NEVR – for data sets that should never be managed

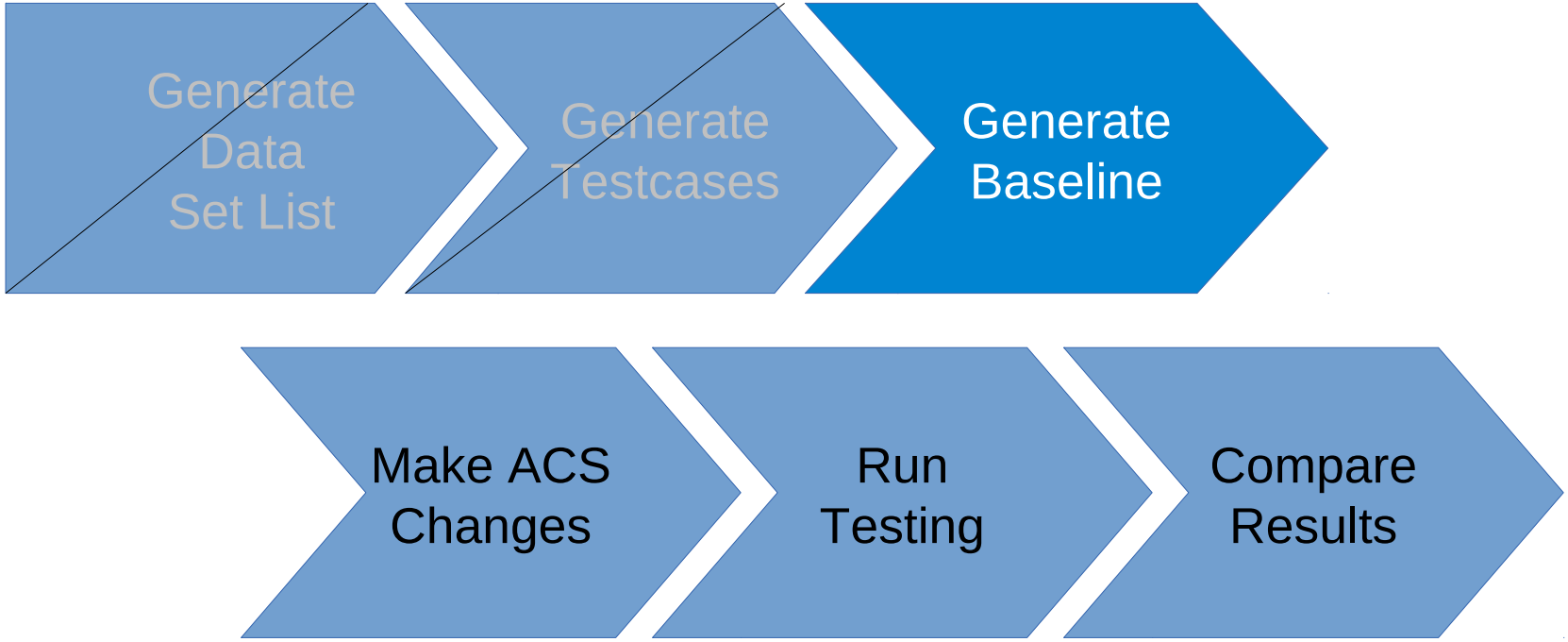
ACS Testing – Generating Testcases

```
  Menu  Functions  Utilities  Help
 ------------------------------------------------------------------------
 BROWSE    NEAL.SMS.ACSTST                                  Row 00001 of 00007
    Name     Prompt          Size    Created          Changed            ID
 . NB1
 . NB2
 . NB3
 . NB4
 . NB5
 . NB6
 . NB7
   **End**



 Command ===> ____________________________________________ Scroll ===> PAGE
  F1=Help    F2=Split   F3=Exit    F5=Rfind   F7=Up      F8=Down    F9=Swap
 F10=Left   F11=Right  F12=Cancel
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  Menu  Utilities  Compilers  Help
 ------------------------------------------------------------------------
 BROWSE    NEAL.SMS.ACSTST(NB1)                       Line 00000000 Col 001 080
********************************** Top of Data **********************************
DESCRIPTION1:
TEST CASE CREATED 2013/08/09 AT 13:11 BY USER2
DSN: NB.RLS.TEST1.DATA
DSTYPE: PERM
ACSENVIR: ALLOC
MAXSIZE: 13280
DSORG: VS
SIZE: 830
NVOL: 1
VOL: 01
XP0403
APPLIC: CICSJOB
UNIT: 3390
********************************* Bottom of Data ********************************
```

ACS Testing - Process



ACS Testing – Generate Baseline

- Simply run the tests and save the output
- ISMF 7.4.3 (ACS routines, Test ACS Routines, Test)

```
   Panel  Utilities  Help

                              TEST ACS ROUTINES

 To Perform ACS Testing, Specify:

   CDS Name . . . . . . 'NEAL.SMS.SCDS'
                        (1 to 44 Character Data Set Name or 'Active')
   ACS Test Library . . 'NEAL.SMS.ACSTST'
   ACS Test Member  . . NB*      (fully or partially specified or * for all
                        members)
   Listing Data Set . . 'NEAL.SMS.BASELINE'
                        (1 to 44 Character Data Set Name or Blank)

 Select which ACS Routines to Test:

   DC ===> Y  (Y/N)  SC ===> Y  (Y/N)  MC ===> Y  (Y/N)  SG ===> Y  (Y/N)
```

ACS Testing – Generate Baseline

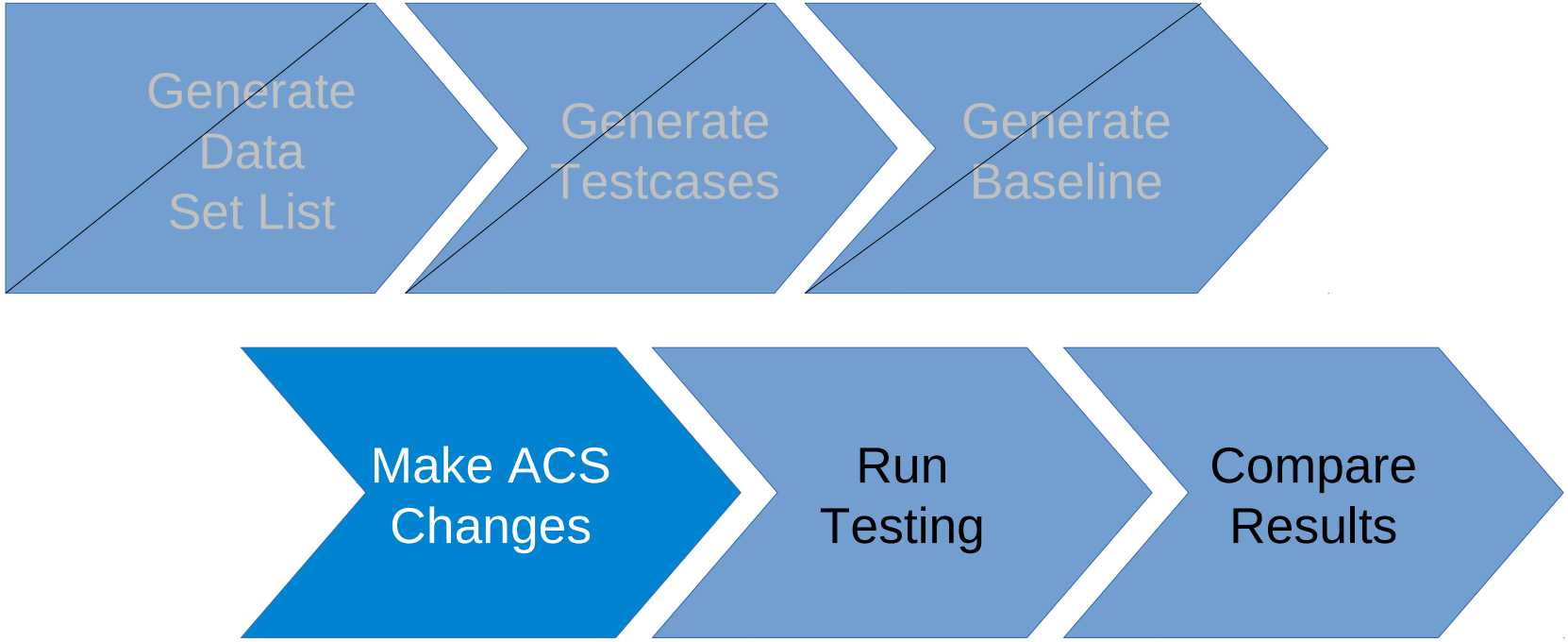
```
 BROWSE    NEAL.SMS.BASELINE
****************************************************************** Top of Data ************
                                ACS TESTING RESULTS

CDS NAME          : NEAL.SMS.SCDS
ACS ROUTINE TYPES: DC SC MC SG
ACS TEST LIBRARY : NEAL.SMS.ACSTST

  ACS TEST
  MEMBER       EXIT CODE   RESULTS
  ---------    ----------  ------------------------------------
  DESCRIPTION: TEST CASE CREATED 2013/08/09 AT 13:11 BY USER2
  EXPECTED RESULT:
  NB1                    0  DC = RLS
    MSG : DATACLAS=RLS
                         0  SC = RLS
    MSG : STORCLAS=RLS
                         0  MC = NULL VALUE ASSIGNED
                         0  SG = SG1

  DESCRIPTION: TEST CASE CREATED 2013/08/09 AT 13:11 BY USER2
  EXPECTED RESULT:
  NB2                    0  DC = RLS
    MSG : DATACLAS=RLS
                         0  SC = RLS
    MSG : STORCLAS=RLS
                         0  MC = NULL VALUE ASSIGNED
                         0  SG = SG1
```

ACS Testing - Process

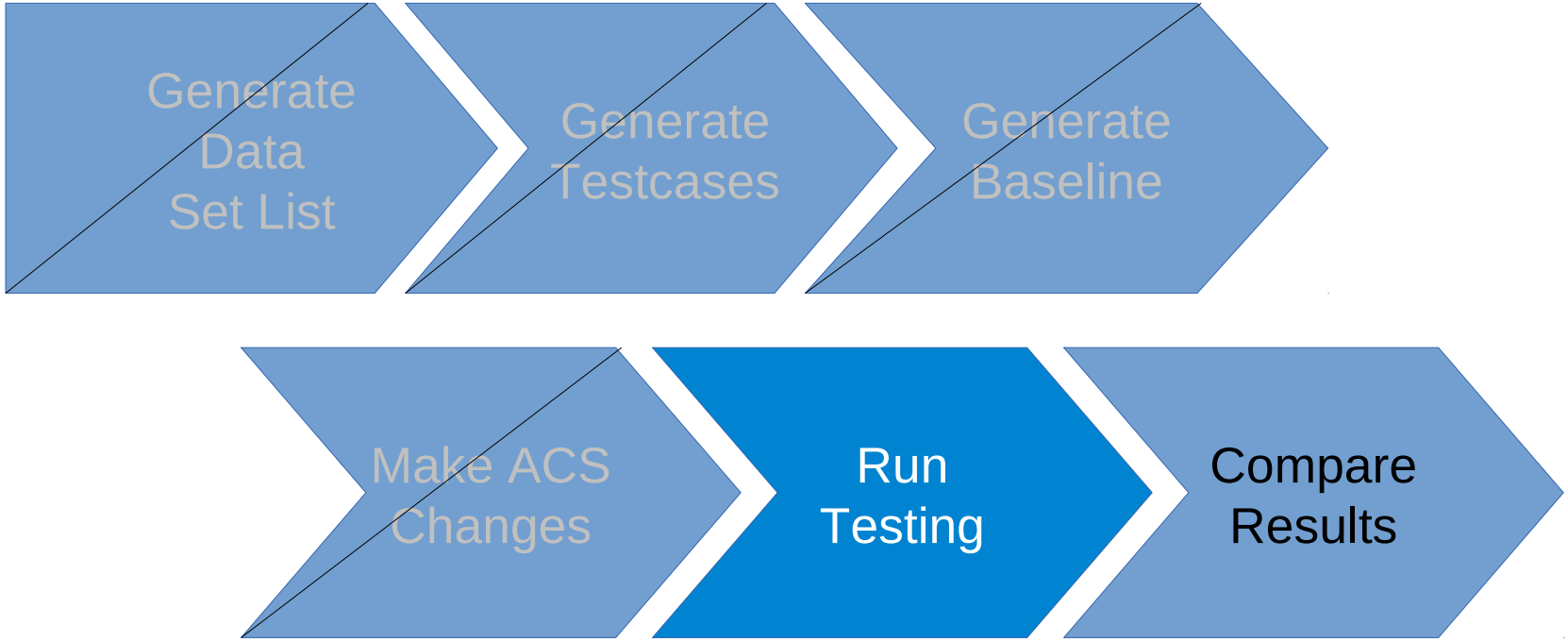


Make Changes to ACS

- Make changes to your ACS
- Recommended that you make changes in a COPY
- Or keep a backup

- Remember to translate and validate!
  - ISMF 7.2 and 3
  - Can translate into a testing SCDS if you prefer

ACS Testing - Process



ACS Testing – Run tests

- Done in the same way as building baseline:
- ISMF 7.4
- Save to a new listing DS

```
                              TEST ACS ROUTINES

To Perform ACS Testing, Specify:

  CDS Name . . . . . . 'NEAL.SMS.SCDS'
                        (1 to 44 Character Data Set Name or 'Active')
  ACS Test Library . . 'NEAL.SMS.ACSTST'
  ACS Test Member  . . NB*      (fully or partially specified or * for all
                        members)
  Listing Data Set . . 'NEAL.SMS.NEWCFG'
                        (1 to 44 Character Data Set Name or Blank)

Select which ACS Routines to Test:

  DC ===> Y  (Y/N)  SC ===> Y  (Y/N)  MC ===> Y  (Y/N)  SG ===> Y  (Y/N)



Use ENTER to Perform Verification and Testing;
Use HELP Command for Help; Use END Command to Exit.
```

ACS Testing – Run Tests Again

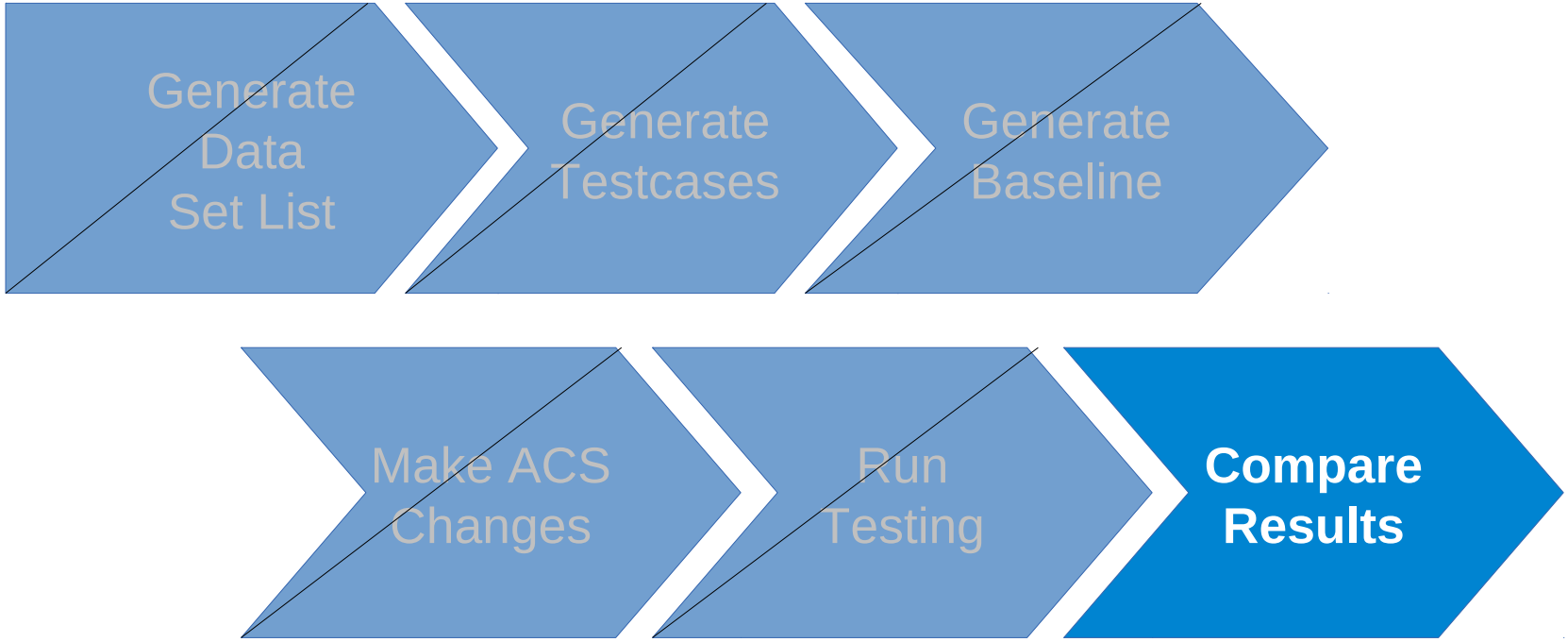
```
 BROWSE    NEAL.SMS.NEWCFG
****************************************************************** Top of Data *****
                                ACS TESTING RESULTS

CDS NAME          : NEAL.SMS.SCDS
ACS ROUTINE TYPES: DC SC MC SG
ACS TEST LIBRARY : NEAL.SMS.ACSTST

  ACS TEST
  MEMBER       EXIT CODE   RESULTS
  ---------    ----------  ------------------------------------
  DESCRIPTION: TEST CASE CREATED 2013/08/09 AT 13:11 BY USER2
  EXPECTED RESULT:
  NB1                    0  DC = RLS
    MSG : DATACLAS=RLS
                         0  SC = SMS
    MSG : STORCLAS=SMS
                         0  MC = NULL VALUE ASSIGNED
                         0  SG = SG1

  DESCRIPTION: TEST CASE CREATED 2013/08/09 AT 13:11 BY USER2
  EXPECTED RESULT:
  NB2                    0  DC = RLS
    MSG : DATACLAS=RLS
                         0  SC = SMS
    MSG : STORCLAS=SMS
                         0  MC = NULL VALUE ASSIGNED
                         0  SG = SG1
```

ACS Testing - Process



ACS Testing – Compare Results

- ISMF Option 11.2
  (NaviQuest ACS Testing Listing Comparison)

```
                    ACS TEST LISTINGS COMPARISON ENTRY PANEL

 To compare ACS listings, specify the following information and press Enter:
   Input Data Sets:
     Base ACS Test Listing (Before latest ACS routine changes)
       ===> 'NEAL.SMS.BASELINE'
     New ACS Test Listing (After latest ACS routine changes)
       ===> 'NEAL.SMS.NEWCFG'
   Reference Data Set for Compare:
     Test Case PDS (Test source for listings above)
       ===> 'NEAL.SMS.ACSTST'

   Output Data Sets:
     Comparison Results Data Set (Summary of exception test cases)
       ===> 'NEAL.SMS.EXPT'
       Replace Contents if DSN Exists  . . Y  (Y or N)
     Exception Test Case PDS (Contents of exception test cases)
       ===> 'NEAL.SMS.EXPTC'
       Replace Contents if DSN Exists  . . Y  (Y or N)

 Use HELP Command for Help; Use END Command to Exit.
```

ACS Testing – Compare Results

```
 BROWSE    NEAL.SMS.EXPT
*********************************************************** Top of Data ****************
                 ACS TEST LISTINGS COMPARISON REPORT

Base ACS listing   :   NEAL.SMS.BASELINE
New ACS listing    :   NEAL.SMS.NEWCFG
Testcase dataset   :   NEAL.SMS.ACSTST
Exception dataset  :   NEAL.SMS.EXPTC


TESTCASE MEMBER : NB1
                       * * * * E X C E P T I O N * * * * *
             RC DATACLAS      RC STORCLAS      RC MGMTCLAS      RC STORGROP
             -- --------      -- --------      -- --------      -- --------
     BASE     0 RLS            0 RLS            0 NULL           0 SG1
      NEW     0 RLS            0 SMS            0 NULL           0 SG1

                      DSN: NB.RLS.TEST1.DATA
                      SIZE: 830
                      VOL: XP0403
                      UNIT: 3390

TESTCASE MEMBER : NB2
                       * * * * E X C E P T I O N * * * * *
             RC DATACLAS      RC STORCLAS      RC MGMTCLAS      RC STORGROP
             -- --------      -- --------      -- --------      -- --------
     BASE     0 RLS            0 RLS            0 NULL           0 SG1
      NEW     0 RLS            0 SMS            0 NULL           0 SG1

                      DSN: NB.RLS.TEST1.INDEX
                      SIZE: 55
                      VOL: XP0403
                      UNIT: 3390
```

ACS Testing - Summary

- Build test cases from data set lists
- Develop a suite of test cases (different prefixes)
- Run the entire suite on any changes – verify results

- ISMF 11.4 – Update test cases with new expected results
  - Updates test cases so that they know what to expect

Stretch break

NaviQuest Batch

- Same functions, different interface
- Plenty of sample JCL exists – start there
  - Update the JOBCARD
  - Update everything after the SYSTSIN DD *
- Controlled by parameters of format PARM(VALUE)
  - Well-documented in JCL Comments
  - Also documented in *DFSMSdfp Storage Administration*
- CLISTS and JCL stored in SYS1.SACBCNTL
  - EXEC ACBJBAOB
  - Specific CLIST called via ISPSTART CMD(cmd name)

ACS Testing in Batch

- ISMF 11.7 Gives Sample JCL for batch processing
- ACS Testing is in 11.7.3 – Configuration Changes Batch

```
                CONFIGURATION CHANGES BATCH SAMPLES SELECTION MENU

Select an option by typing '/' or enter Data Set to Edit and press Enter:

                                                  More:    -
    _  Validate SCDS
    _  Test ACS Routines
    _  Generate Enhanced ACS Test Listing
    _  Compare ACS Test Listings
    _  Update Test Cases with Test Results
    _  Delete Management Class
    _  Delete Data Class
    _  Delete Storage Class
    _  Delete Storage Group
    _  Delete Copy Pool
    _  Delete Aggregate Group
    _  Delete Tape Library

Data Set to Edit . . _____________________________________________________

Use HELP Command for Help; Use END Command to Exit.
```

ACS Testing in Batch - Testing

```
//*****************************************************************
//*                                                               *
//*   TEST STEP                                                   *
//*                                                               *
//*    SCDS     - NAME OF SCDS THAT CONTAINS THE TRANSLATED,     @D1C*
//*               VALIDATED ACS ROUTINES TO BE TESTED (INPUT)     *
//*    TESTBED  - PDS CONTAINING TEST CASES THAT THE ACS ROUTINES    *
//*               SHOULD BE TESTED FOR (INPUT)                    *
//*    MEMBER   - MEMBERS TO BE TESTED IN TESTBED (INPUT)         *
//*    DC,SC,MC,SG - ROUTINES TO BE TESTED Y OR N (INPUT)         *
//*    LISTNAME - TEST LISTING (OUTPUT)                           *
//*                                                               *
//*****************************************************************
//*****************************************************************
//TESTACS  EXEC ACBJBAOB,
//         PLIB1='SYS1.DGTPLIB',
//         TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
DEL NEW.TESTLIST
ISPSTART CMD(ACBQBAIA +
SCDS(MYSCDS) +
TESTBED(TESTCASE.LIBRARY) MEMBER(*) +
LISTNAME(NEW.TESTLIST) +
DC(Y) SC(Y) MC(Y) SG(Y)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
```

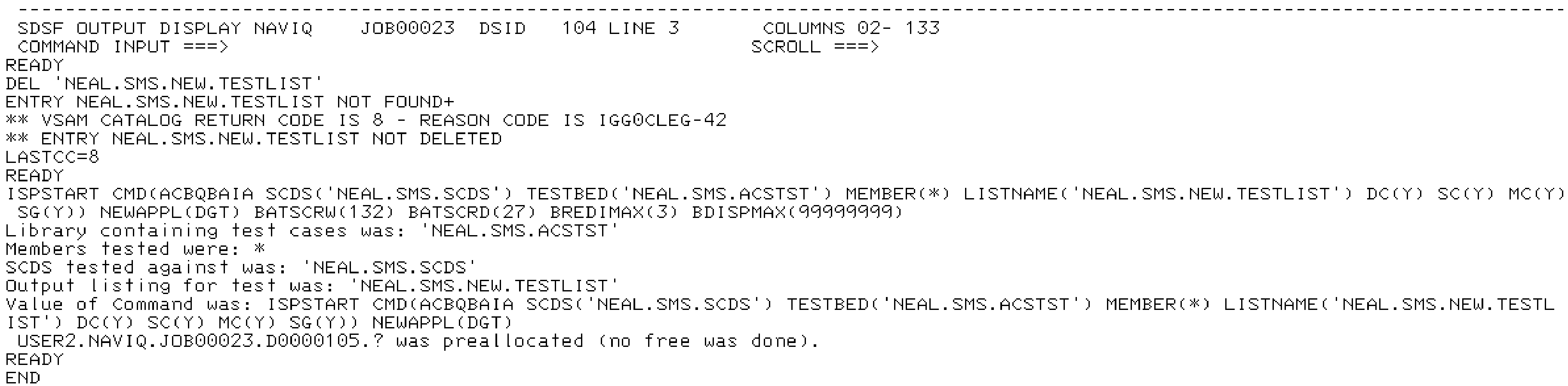
ACS Testing in Batch - Testing

```
//*****************************************************************
//*                                                               *
//*   TEST STEP                                                   *
//*                                                               *
//*    SCDS     - NAME OF SCDS THAT CONTAINS THE TRANSLATED,     @D1C*
//*               VALIDATED ACS ROUTINES TO BE TESTED (INPUT)     *
//*    TESTBED  - PDS CONTAINING TEST CASES THAT THE ACS ROUTINES  *
//*               SHOULD BE TESTED FOR (INPUT)                    *
//*    MEMBER   - MEMBERS TO BE TESTED IN TESTBED (INPUT)         *
//*    DC,SC,MC,SG - ROUTINES TO BE TESTED Y OR N (INPUT)         *
//*    LISTNAME - TEST LISTING (OUTPUT)                           *
//*                                                               *
//*****************************************************************
//*****************************************************************
//TESTACS  EXEC ACBJBAOB,
//         PLIB1='SYS1.DGTPLIB',
//         TABL2=NEAL.SMS.ISPTABL
//SYSTSIN  DD *
PROFILE PREFIX(USER2)
DEL 'NEAL.SMS.NEW.TESTLIST'
ISPSTART CMD(ACBQBAIA +
SCDS('NEAL.SMS.SCDS') +
TESTBED('NEAL.SMS.ACSTST') MEMBER(*) +
LISTNAME('NEAL.SMS.NEW.TESTLIST') +
DC(Y) SC(Y) MC(Y) SG(Y)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
```

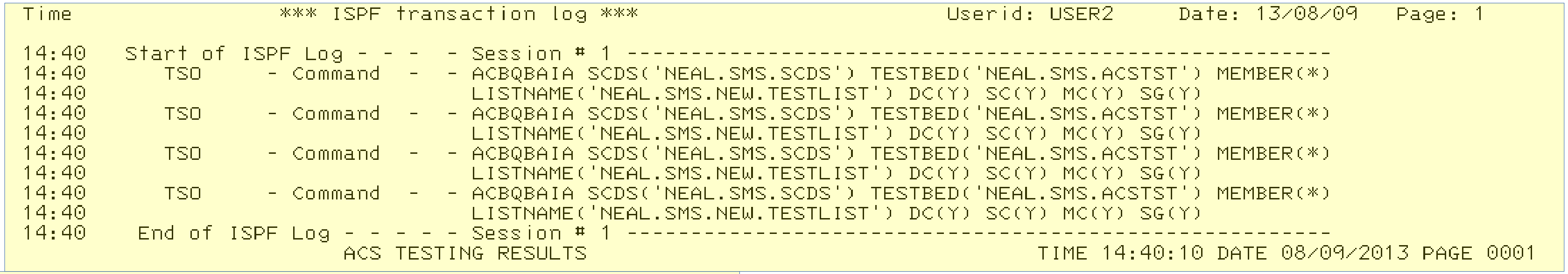
TABL2 must be allocated
It is a standard ISPF table.

PDS Dataset
RECFM=FB
LRECL=80
DSORG=PO (PDS)

ACS Testing in Batch - Results

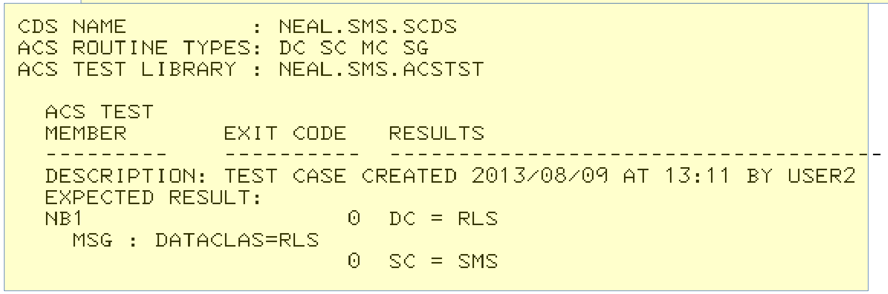


```
 ------------------------------------------------------------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY NAVIQ    JOB00023  DSID   104 LINE 3       COLUMNS 02- 133
 COMMAND INPUT ===>                                            SCROLL ===>
READY
DEL 'NEAL.SMS.NEW.TESTLIST'
ENTRY NEAL.SMS.NEW.TESTLIST NOT FOUND+
** VSAM CATALOG RETURN CODE IS 8 - REASON CODE IS IGG0CLEG-42
** ENTRY NEAL.SMS.NEW.TESTLIST NOT DELETED
LASTCC=8
READY
ISPSTART CMD(ACBQBAIA SCDS('NEAL.SMS.SCDS') TESTBED('NEAL.SMS.ACSTST') MEMBER(*) LISTNAME('NEAL.SMS.NEW.TESTLIST') DC(Y) SC(Y) MC(Y)
 SG(Y)) NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
Library containing test cases was: 'NEAL.SMS.ACSTST'
Members tested were: *
SCDS tested against was: 'NEAL.SMS.SCDS'
Output listing for test was: 'NEAL.SMS.NEW.TESTLIST'
Value of Command was: ISPSTART CMD(ACBQBAIA SCDS('NEAL.SMS.SCDS') TESTBED('NEAL.SMS.ACSTST') MEMBER(*) LISTNAME('NEAL.SMS.NEW.TESTL
IST') DC(Y) SC(Y) MC(Y) SG(Y)) NEWAPPL(DGT)
 USER2.NAVIQ.JOB00023.D0000105.? was preallocated (no free was done).
READY
END
```



```
 Time                *** ISPF transaction log ***                          Userid: USER2     Date: 13/08/09    Page: 1

 14:40   Start of ISPF Log - - -  - Session # 1 -------------------------------------------------------
 14:40      TSO     - Command   -  - ACBQBAIA SCDS('NEAL.SMS.SCDS') TESTBED('NEAL.SMS.ACSTST') MEMBER(*)
 14:40                               LISTNAME('NEAL.SMS.NEW.TESTLIST') DC(Y) SC(Y) MC(Y) SG(Y)
 14:40      TSO     - Command   -  - ACBQBAIA SCDS('NEAL.SMS.SCDS') TESTBED('NEAL.SMS.ACSTST') MEMBER(*)
 14:40                               LISTNAME('NEAL.SMS.NEW.TESTLIST') DC(Y) SC(Y) MC(Y) SG(Y)
 14:40      TSO     - Command   -  - ACBQBAIA SCDS('NEAL.SMS.SCDS') TESTBED('NEAL.SMS.ACSTST') MEMBER(*)
 14:40                               LISTNAME('NEAL.SMS.NEW.TESTLIST') DC(Y) SC(Y) MC(Y) SG(Y)
 14:40      TSO     - Command   -  - ACBQBAIA SCDS('NEAL.SMS.SCDS') TESTBED('NEAL.SMS.ACSTST') MEMBER(*)
 14:40                               LISTNAME('NEAL.SMS.NEW.TESTLIST') DC(Y) SC(Y) MC(Y) SG(Y)
 14:40    End of ISPF Log - - - - - Session # 1 -------------------------------------------------------
                         ACS TESTING RESULTS                                    TIME 14:40:10 DATE 08/09/2013 PAGE 0001
```



```
CDS NAME         : NEAL.SMS.SCDS
ACS ROUTINE TYPES: DC SC MC SG
ACS TEST LIBRARY : NEAL.SMS.ACSTST

  ACS TEST
  MEMBER       EXIT CODE   RESULTS
  ---------    ----------  -------------------------------------
  DESCRIPTION: TEST CASE CREATED 2013/08/09 AT 13:11 BY USER2
  EXPECTED RESULT:
  NB1                   0  DC = RLS
    MSG : DATACLAS=RLS
                        0  SC = SMS
```

ACS Testing in Batch – Compare job

```
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO COMPARE ACS TEST LISTINGS IN BATCH                *
//*                                                                 *
//*   INSTRUCTIONS BEFORE SUBMITTING:                               *
//*                                                                 *
//*     CHANGE JOBCARD                                              *
//*     CHANGE PREFIX                                               *
//*     CHANGE PARAMETERS                                           *
//*                                                                 *
//*   PARAMETERS:                                                   *
//*                                                                 *
//*     BASELIST - BASE ACS TEST LISTING (INPUT)                    *
//*     NEWLIST  - NEW ACS TEST LISTING (INPUT)                     *
//*     TESTBED  - TEST CASE PDS (REFERENCE INPUT)                  *
//*     RSLTDSN  - COMPARISON RESULTS DATA SET (OUTPUT)             *
//*     XCPTPDS  - EXCEPTION TEST CASE PDS (OUTPUT)                 *
//*     XCPSPACE - SPACE values of Except DS (Optional)    3a@WA32832*
//*       Values: (Primary Tracks,Secondary Tracks,Directory Blocks) *
//*         which are positional and optional. Defaults: (3,1,20).  *
//*                                                                 *
//*   NOTE:  If you receive message IEC217I B14-0C on your exception *
//*       data set, you need to increase your data set size by using *
//*       the XCSPACE parameter (specially the directory blocks)    *
//*                                                                 *
//*******************************************************************
```

```
//CMPRSTEP EXEC  ACBJBAOB,
//         PLIB1=SYS1.DGTPLIB,
//         TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
DEL COMPARE.LISTING
DEL TESTCASE.EXCP
ISPSTART CMD(%ACBQBAC1 +
BASELIST(BASE.TESTLIST) +
NEWLIST(NEW.TESTLIST) +
TESTBED(TESTCASE.LIBRARY) +
RSLTDSN(COMPARE.LISTING) +
XCPTPDS(TESTCASE.EXCP) +
XCPSPACE(5,3,30)) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3)
BDISPMAX(99999999)
/*
```

ACS Batch Summary

- Sample JCL is in ISMF 11.7 or SYS1.SACBCNTL
- Change the necessary parameters
- Submit

- Get work done.

Sources

- *DFSMS Implementing System-Managed Storage*
  - SC26-7407
  - http://publib.boulder.ibm.com/infocenter/zos/v1r11/topic/com.ibm.zos.r11.idai600/acsrec.htm

- *DFSMSdfp Storage Administration*
  - SC26-7402
  - http://publibfp.dhe.ibm.com/cgi-bin/bookmgr/BOOKS/DGT2S2A1/17.0?DT=20120126150025

#SHAREorg

SHARE

DFSMS:Intermediate
Understanding Someone Else's ACS Routines

Neal Bohling
DFSMS Defect Support, IBM

August 14, 2013
Session# 13158

SHARE in Boston

Notices & Disclaimers

Product information and data has been reviewed for accuracy as of the date of initial publication. Product information and data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the products and/or programs described herein at any time without notice.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Consult your local IBM representative or IBM Business Partner for information about the product and services available in your area.

Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectually property rights, may be used instead. It is the user's responsibility to evaluate and verify the operation of any non-IBM product, program or service.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS"WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. IBM is not responsible for the performance or interoperability of any non-IBM products discussed herein.

Notices & Disclaimers

The performance data contained herein was obtained in a controlled, isolated environment. Actual results that may be obtained in other operating environments may vary significantly. While IBM has reviewed each item for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere.

The responsibility for use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's or user's ability to evaluate and integrate them into their operating environment. Customers or users attempting to adapt these techniques to their own environments do so at their own risk. IN NO EVENT SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not necessarily tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or another claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Trademarks

DFSMSdfp, DFSMSdss, DFSMShsm, DFSMSrmm, IBM, IMS, MVS, MVS/DFP, MVS/ESA, MVS/SP, MVS/XA, OS/390, SANergy, and SP are trademarks of International Business Machines Corporation in the United States, other countries, or both.

AIX, CICS, DB2, DFSMS/MVS, Parallel Sysplex, OS/390, S/390, Seascape, and z/OS are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Domino, Lotus, Lotus Notes, Notes, and SmartSuite are trademarks or registered trademarks of Lotus Development Corporation. Tivoli, TME, Tivoli Enterprise are trademarks of Tivoli Systems Inc. in the United States and/or other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

Other company, product, and service names may be trademarks or service marks of others.