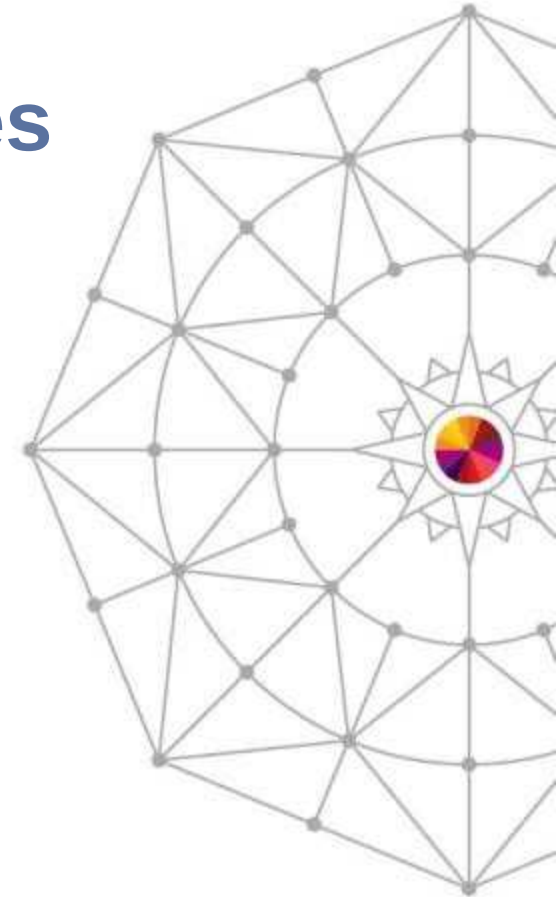


# DFSMS Basics: How to Write ACS Routines Hands-on Lab (Section 1)

Neal Bohling and Tom Reed, IBM

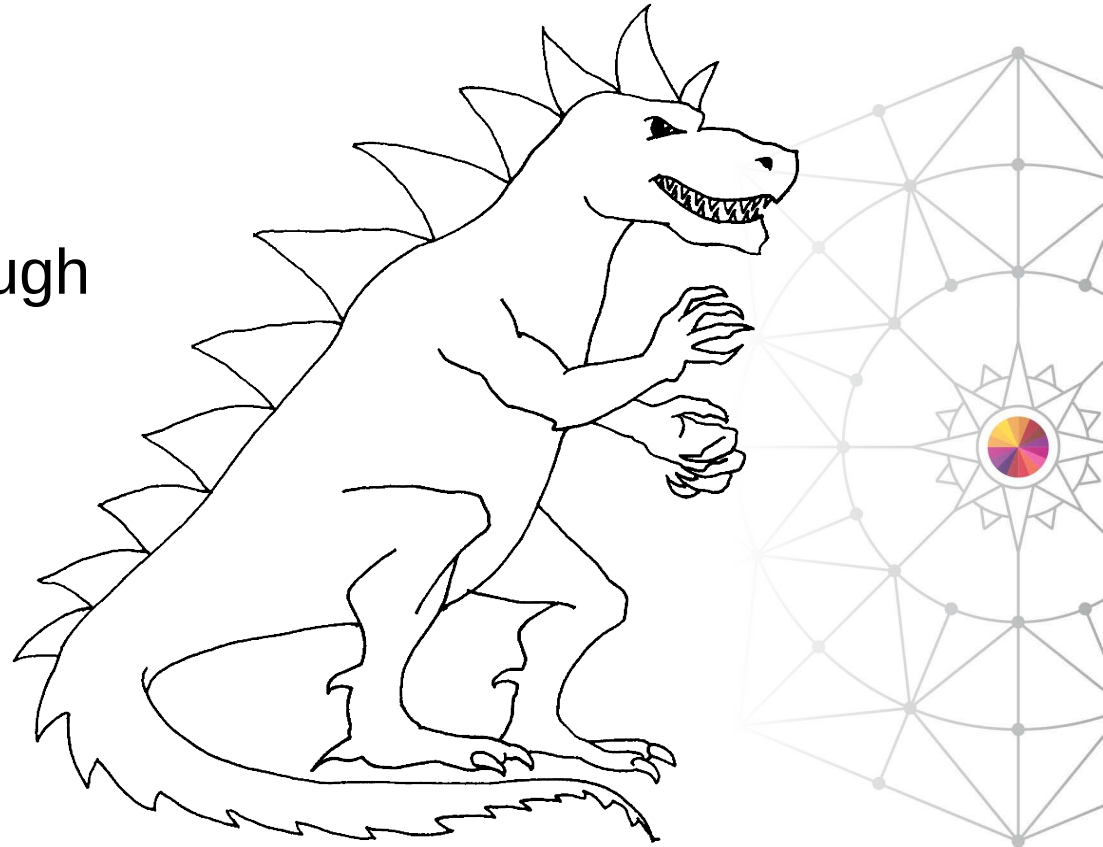
March 12, 2014

Session Number 15097



# Agenda

- Short intro to SMS
- Configuration Walkthrough
- ACS Walkthrough
- Lab



# Introduction to SMS

**S** STORAGE

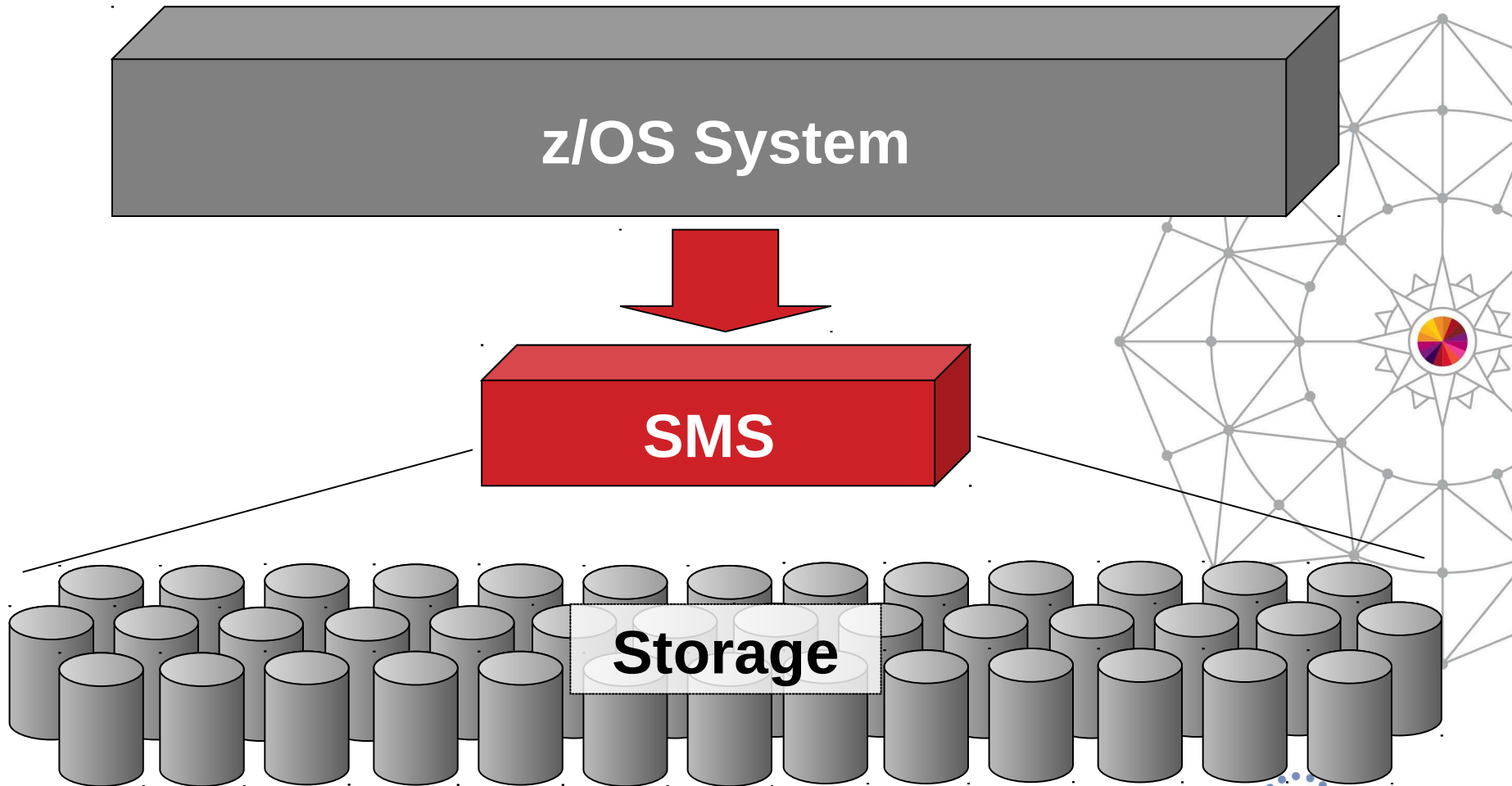
**M** MANAGEMENT

**S** SUBSYSTEM

- DFSMS facility designed for automating and centralizing storage management.
- Allows you to define
  - data allocation characteristics
  - performance and availability goals,
  - backup and retention requirements
  - storage requirements
- Benefits:
  - Improves storage space use
  - Allows central control
  - Enables you to manage storage growth more efficiently

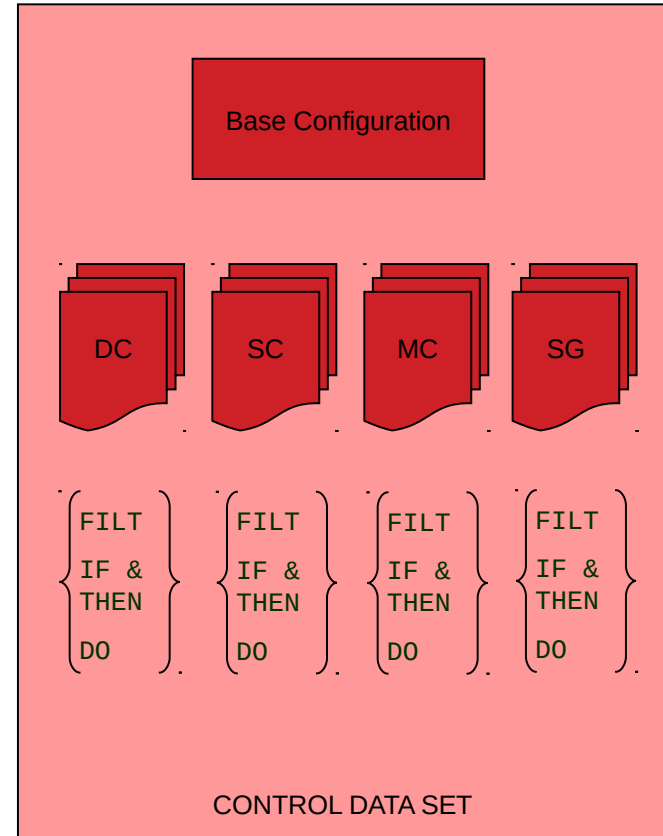


# Introduction to SMS Environment



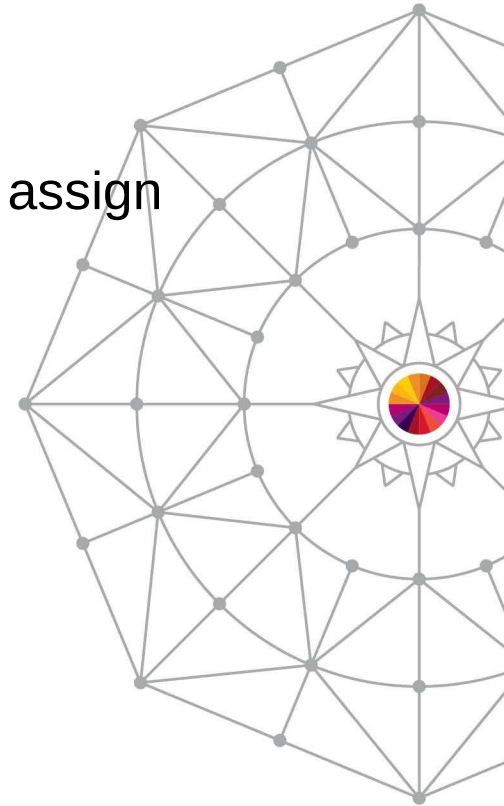
# SMS Configuration Includes

- **Base Configuration**
  - Installation defaults (device geometry)
  - Systems included in the *SMS complex*
  
- **Constructs**
  - Data Classes – basic allocation defaults
  - Storage Classes – access attributes
  - Management Classes – migration information
  - Storage Groups – collection of volumes
  
- **Automating Class Selection (ACS)**
  - User-defined script
  - One per construct
  - Selects construct based on various criteria
  
- **Stored in the Control Data Sets (CDS)**
  - Active CDS – ACDS
  - Source CDS – SCDS
  - Communication CDS - COMMDS

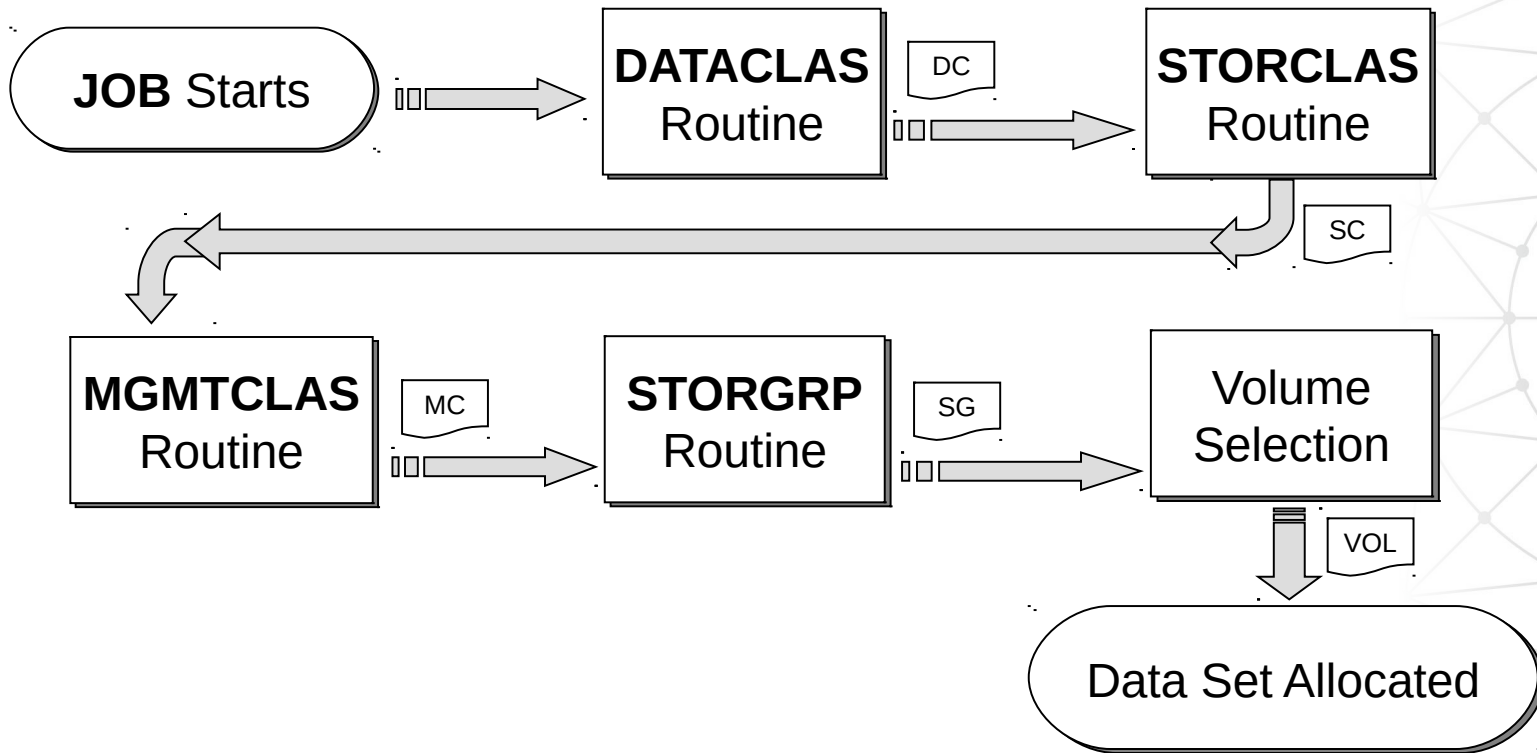


# Introduction to ACS Environment

- **What is an ACS Routine?**
  - User written code
  - Selects which SMS classes and storage groups to assign
  - One per type of construct
  - They run at ALLOCATION time
  - Process in order:
    - DATACLAS
    - STORCLAS
    - MGMTCLAS
    - STORGRP
  - REQUIRED even if nothing is set

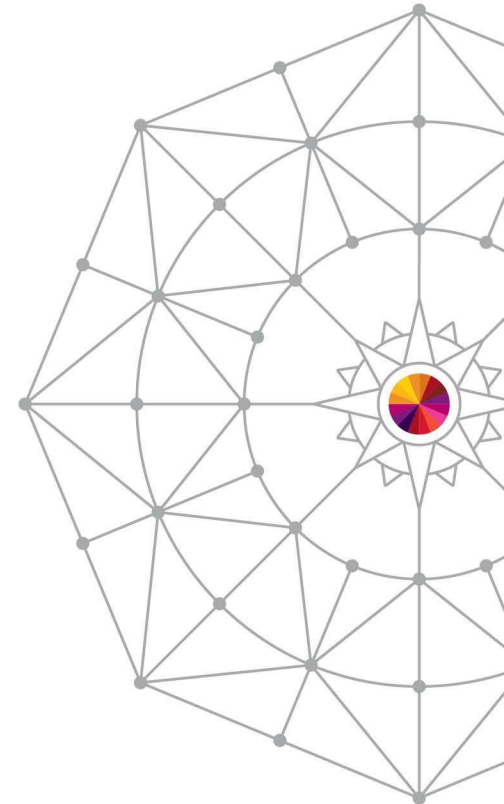


# ACS Routine Process Flow



# Introduction to ACS Environment (cont)

- **ACS Language Statements**
  - **PROC** - beginning of routine
  - **FILTLIST** – defines filter criteria
  - **DO** – start of statement group
  - **SELECT** – defines a set of conditional statements
  - **IF** – conditional statement
  - **SET** – assigns a read/write variable
  - **WRITE** – sends message to end user
  - **EXIT** – immediately terminates ACS routine
  - **END** – end of statement group
  - **/\* COMMENT \*/** - comments a line





# Variables

- Always start with an &
- Two types: READ ONLY, READ/WRITE
- **READ ONLY**
  - 47 different variables
  - Contain data set and system information
  - Reflect what is known at the time of the request
  - Can only be used for comparison
  - Examples: &DSORG, &DSNTYPE, &SIZE, &HLQ
- **READ/WRITE Variables**
  - Used to assign values
  - Only 4 variables
    - &DATACLAS
    - &STORCLAS
    - &MGMTCLAS
    - &STORGRP

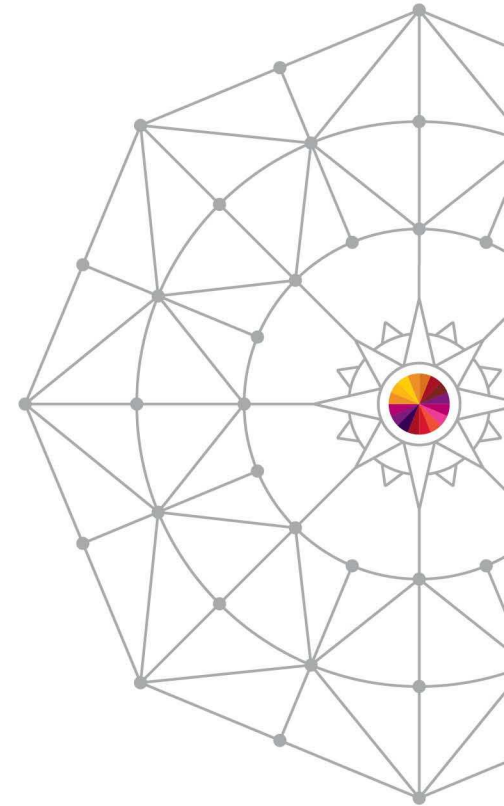


# ACS Standard Flow / Example

```
PROC DATACLAS
/* MY FILTLISTS */
FILTLIST ORIGSTG +
    INCLUDE('LARRY', 'CURLY', MO*) +
    EXCLUDE('SHEMP')

/* LOGIC */
IF( &HLQ EQ &ORIGSTG ) THEN DO
    SET &DATACLAS = 'STGDC'
    END
ELSE SET &DATACLAS = ''

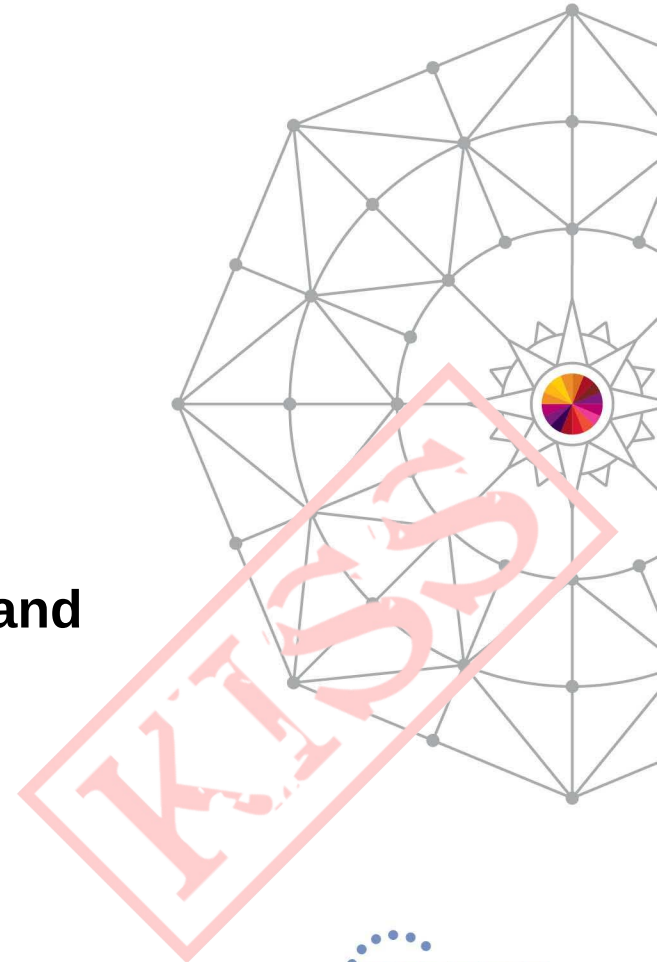
WRITE 'DATACLAS = ' &DATACLAS
END
```



# Introduction to ACS Environment (cont)

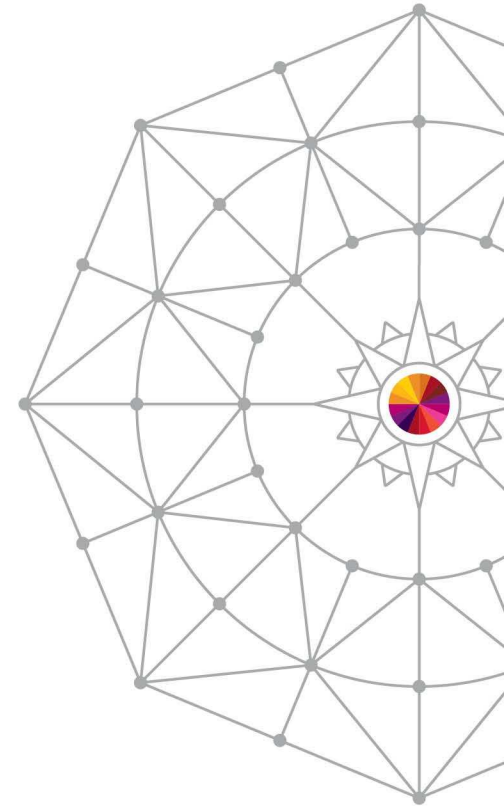
- **ACS General Rules**

- *Know your logic before you code*
- **Keep them simple and straightforward**
  - Minimize exceptions
  - Maximize FILTLIST usage
- **Keep them easy to maintain and understand**
  - Use SELECT instead of IF when possible
  - EXIT the routine as soon as possible
  - Use OTHERWISE whenever possible
  - Comments, comments, comments



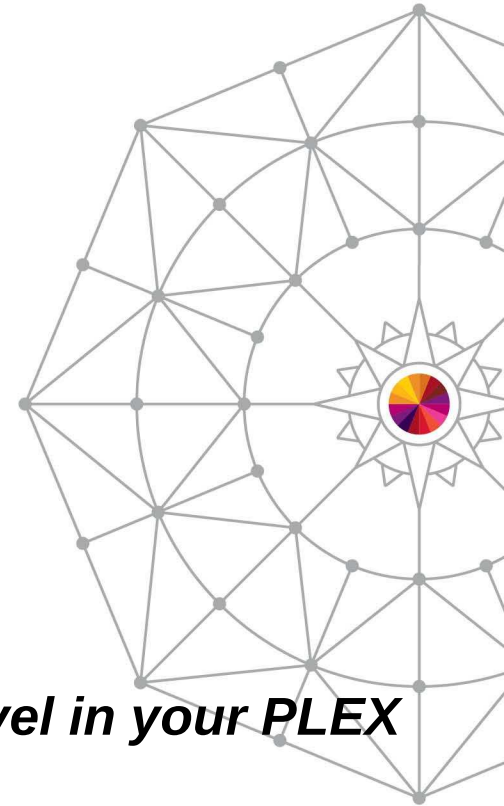
# A Few “Gotchas”

- **Numeric constants are easy: just numbers**
  - *&NQUAL = 5*
- **Suffixes : sizes require KB or MB suffix**
  - *&MAXSIZE = 100MB*
- **String literals are in single quotes**
  - *&HLQ = 'TEST'*
- **Masks are in NOT in quotes**
  - *&DSN = SYS1.\*LIB*
- **&& is AND, | is OR**
- **Watch for fall-through logic in your IF and SELECT**



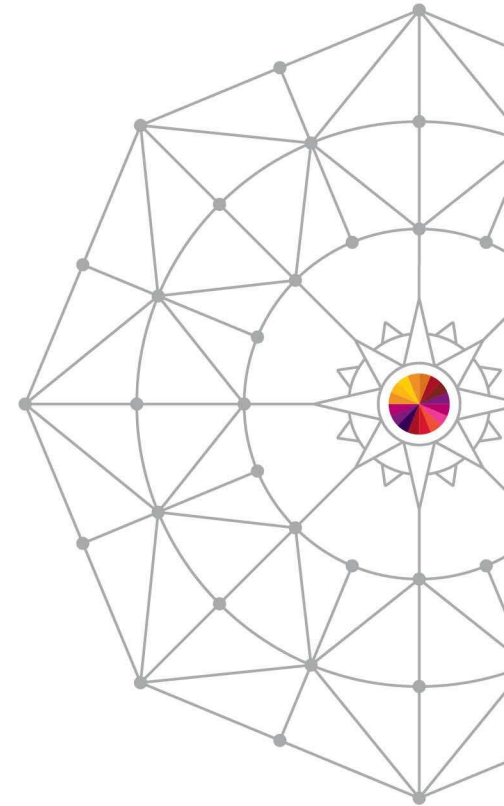
# Introduction to ACS Environment (cont)

- **Write the ACS Routines**
  - Saved in a text format
- **Translate ACS Routines**
  - Converts to byte code and inserts into the SCDS
- **Validate the SMS Configuration**
  - Verifies your construct allocation (do they all exist?)
- **Activate the SMS Configuration**
- ***Note: translate / validate from the highest z/OS level in your PLEX***



# Lab Time

See your handout and start the lab!

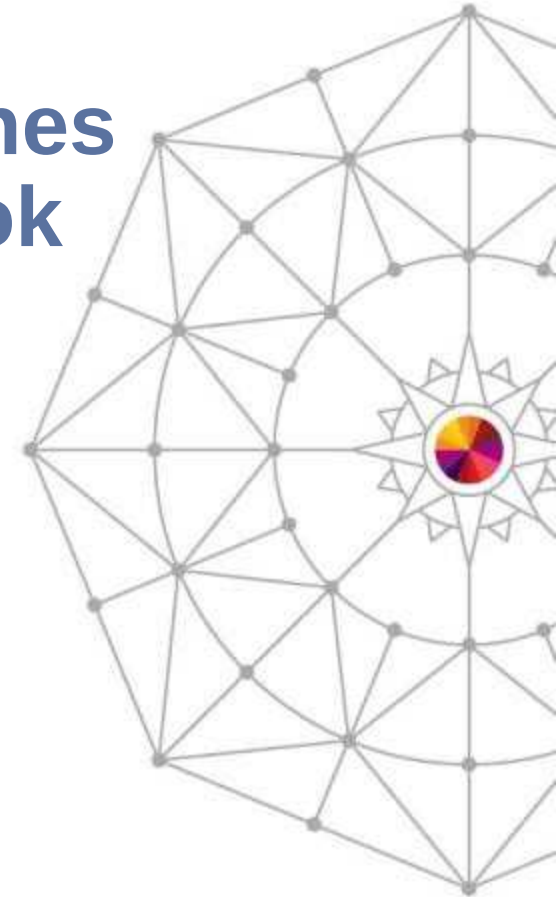


# DFSMS Basics: How to Write ACS Routines Hands-on Lab / Workbook

Neal Bohling and Tom Reed, IBM

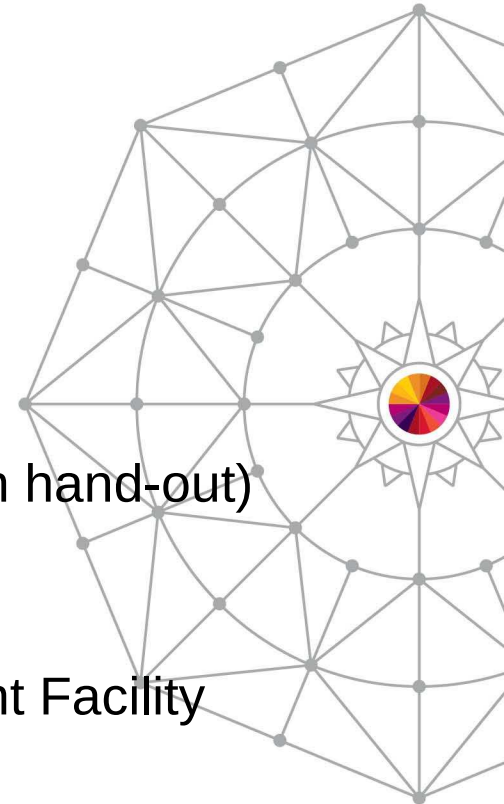
March 12, 2014

Session 15097



# Resources

- **Lab TSO USERIDs**
  - SHARA01 thru SHARA20 w/ password (firstpw)
- **Lab Data Sets**
  - SHARAx.x.S15097.ACS
  - SHARAx.x.S15097.SMS.SCDS
- **Publications**
  - DFSMS Storage Administration Reference (section hand-out)
    - Link: <https://ibm.biz/BdRmgJ>
  - DFSMS Implementing System-Managed Storage
  - DFSMS Using the Interactive Storage Management Facility





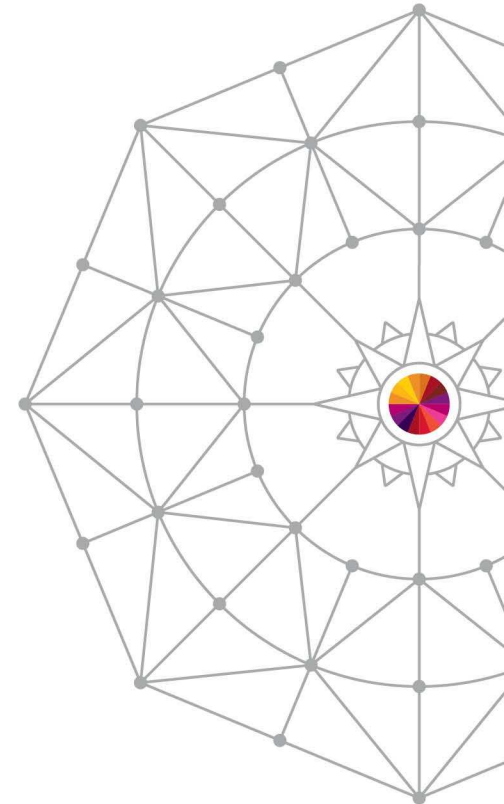
# Lab Basics

- **SHARAx.x.S15097.ACS**
  - Members for your lab use
    - DCLAB
    - SCLAB
    - MCLAB
    - SGLAB
  - Contains the ACS routines for a minimal configuration
    - SCMIN and SGMIN
  - Contains some example routines
    - DCEXAMPL, SCEXAMPL, MCEXAMPL and SGEXAMPL
  - Example solutions
    - DCANSWER, SCANSWER , MCANSWER and SGANSWER
  - ACS Routine stubs
    - STUB
- **SHARAx.x.S15097.SMS.SCDS**
  - SMS Configuration to code ACS against



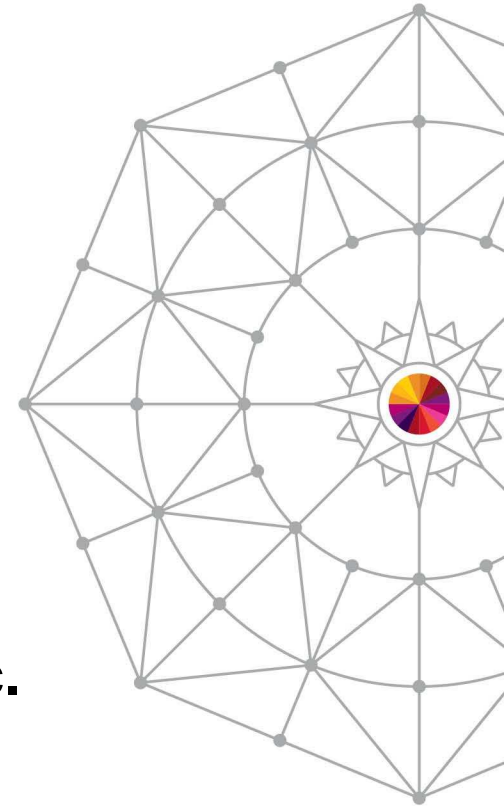
# Lab Configuration

- **Available SCDS:**
  - **SHARAx.x.S15097.SMS.SCDS**
- **Available DATACLAS(s)**
  - Default, Extended, HFS, PDSE, VEAEXTND
- **Available STORCLAS(s)**
  - Default, Extended, GSPACE
- **Available MGMTCLAS(s)**
  - RLSEIMM
- **Available STORGRP(s)**
  - Default, Extended



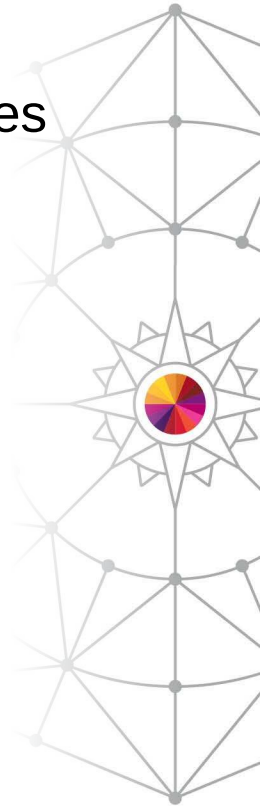
# Lab Pre-stuff

- Set yourself up as a Storage Administrator
  - ISMF
  - 0 - Profile Options
  - 0 - User Mode Selections
  - 2 - Storage Administrator
  - End/Exit 3 times
  
- **NOTE: This is an exercise in syntax, not logic.**



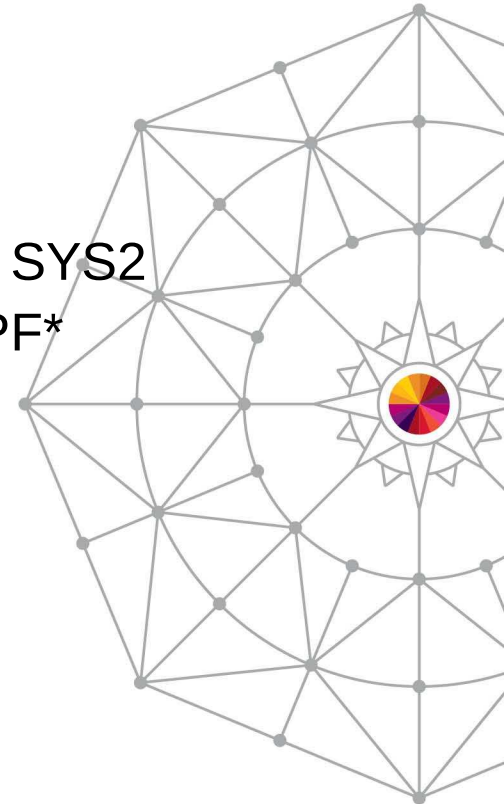
# Lab 1

- **Create the ‘stubs’ for each Class and Group**
- Create the basic entries for each of DC, SC, MC, and SG routines
  - Put them in data set SHARAx.x.S15097.ACS as members:
    - DCLAB
    - SCLAB
    - MCLAB
    - SGLAB
  - The ‘stubs’ can be found in SHARAx.x.S15097.ACS(STUB)
- Translate the ACS routines via ISMF 7 / 2 to your configuration (SCDS) data set (SHARAx.x.S15097.SMS.SCDS)



# Lab 2

- **Create the Filter Lists**
  - Use the SC ACS routine created in Lab 1
  - Modify the routine such that it contains FILTLISTS
    - Create a filter of SYSTEM for HLQ of SYS1 and SYS2
    - Create a filter for SPF for 2<sup>nd</sup> level qualifier of SPF\*
- Translate the ACS routine



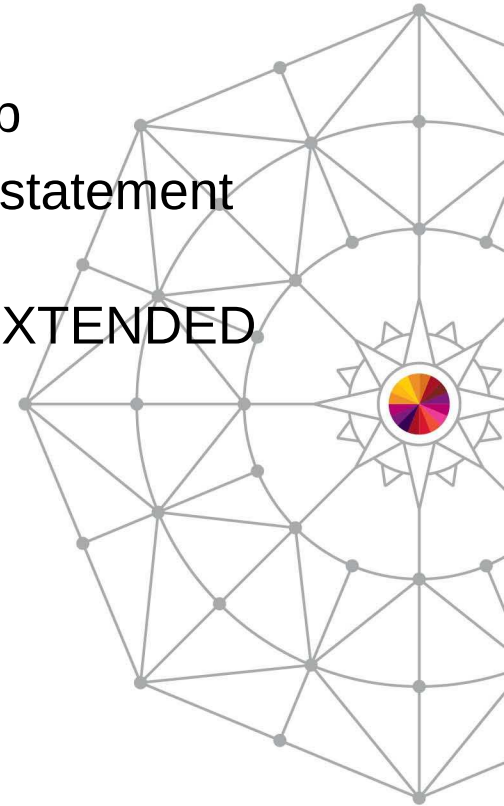
# Lab 3

- **Add some If/Then Logic**
  - Use the SC ACS routine created in the previous lab
  - Modify the routine such that it contains IF/THEN logic
    - Compare the HLQ to the SYSTEM filter and set a null (“”) SC
    - Compare the second-level qualifier to the SPF filter and if it matches, set a Storage Class of Default
- Translate the ACS routine



# Lab 4

- **SELECT Logic**
  - Use the SC ACS routine created in the previous lab
  - Modify the routine such that it contains a SELECT statement
    - SELECT on Read/Write variable &DATACLAS
    - When incoming DC is VEAEXTND set the SC to EXTENDED.
  - Translate the ACS routine



# Lab 5

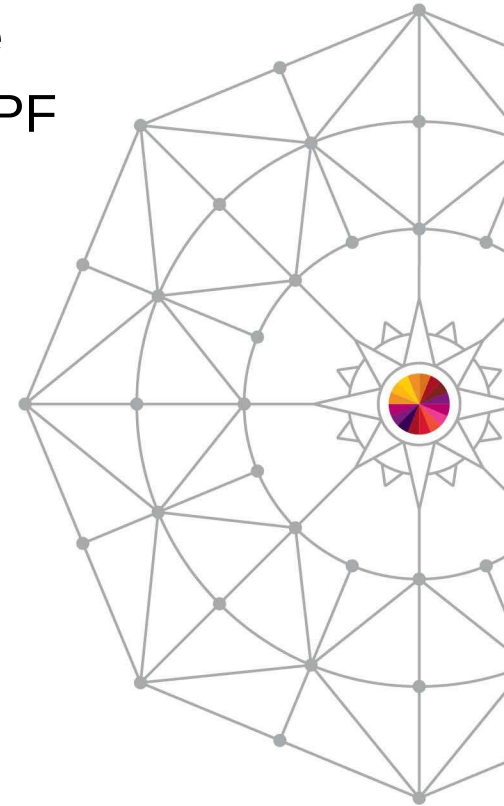
- **WRITE Statement**
  - Use the SC ACS routine created in the previous lab
  - Modify the routine such that it contains a WRITE statement(s) to indicate which storage class is assigned
    - Syntax: WRITE 'message'
    - Try to make sure every exit point has a write statement
  - Translate the ACS routine





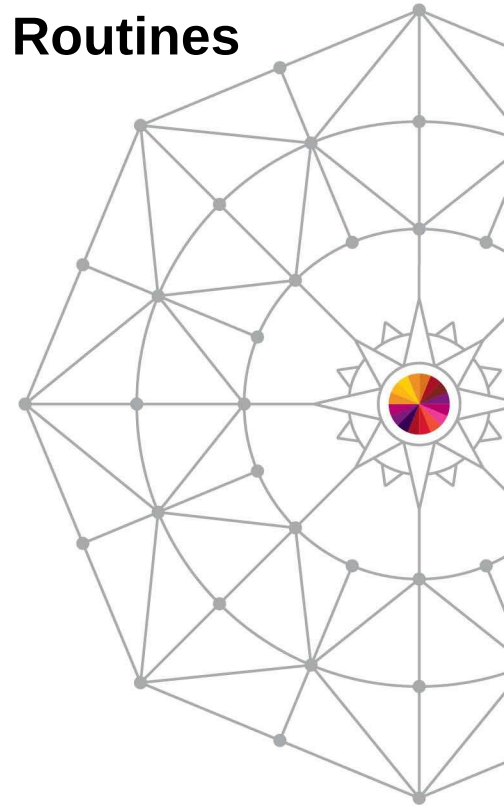
# Lab 6

- **Create another filter list for SPF in the SG Routine**
  - Based on the second-level qualifier starting with SPF
  - Set a Storage Group of Default
- **Translate the ACS routine**



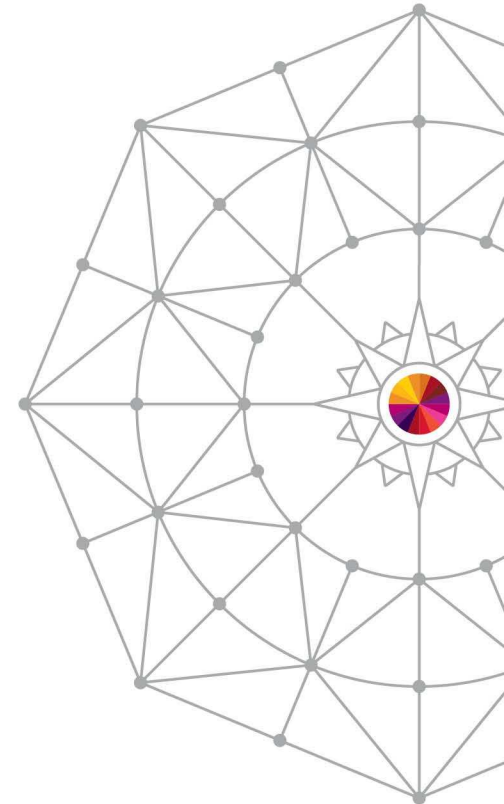
# Lab 7

- **Create another filter list for SPF in the DC and MC Routines**
  - Based on the 2<sup>nd</sup> level qualifier starting with SPF:
    - Set Data Class of Default
    - Management Class of RLSEIMM.
- **Translate the ACS routines**



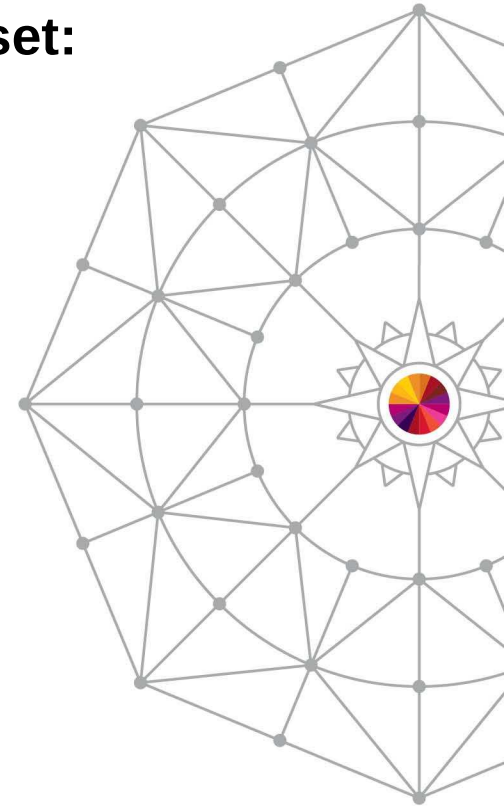
# Lab 8

- **Use SELECT Logic**
  - When the DSNTYPE is LIBRARY:
    - Set a Data Class of PDSE
    - Set a Storage Class of Default, and
    - Set a Storage Group of Default
  - When the DSNTYPE is PDS:
    - Set a Data Class of Default
    - Set a Storage Class of Default
    - Set a Storage Group of Default
- **Translate the ACS routines**



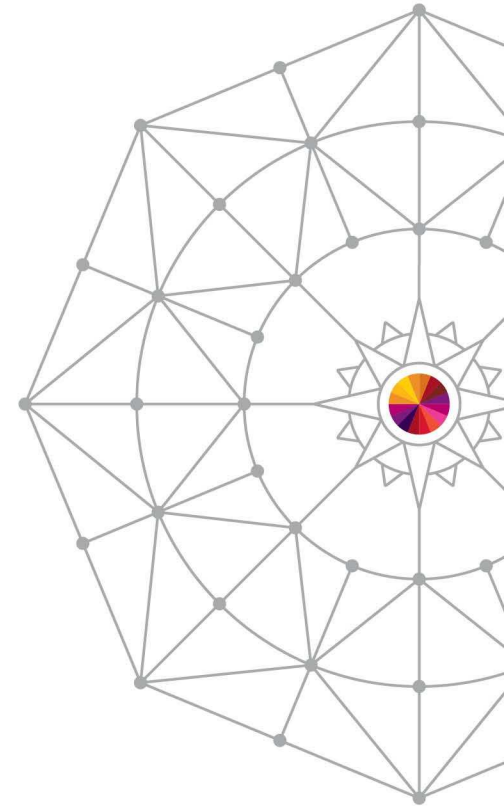
# Lab 9

- **Based on MAXSIZE of 100MB and DSORG of VS, set:**
  - Data Class of 'VEAEXTND',
  - Storage Class of 'Extended'
  - Storage Group of 'Extended'
- HINT: The SC aspect of this assignment is already complete.
- **Translate the ACS routines**



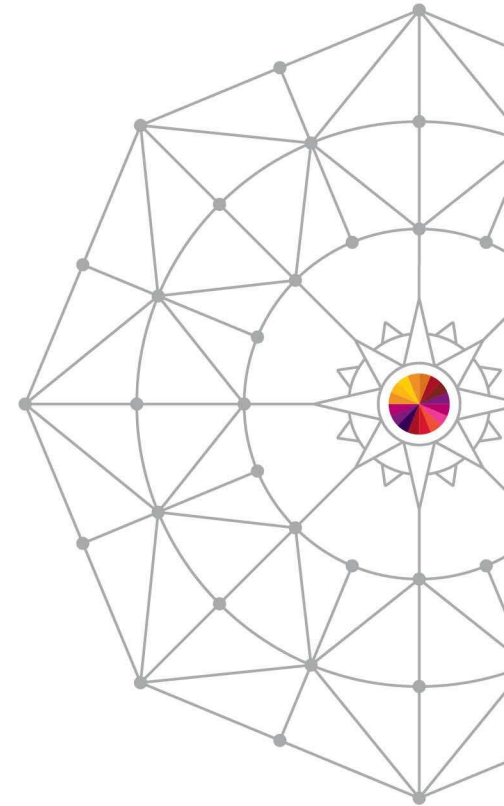
# Lab Assignment (cont)

- **Validate** the configuration
  - ISMF 7.3
  - Use your configuration (SCDS)  
(SHARAx.x.S15097.SMS.SCDS)



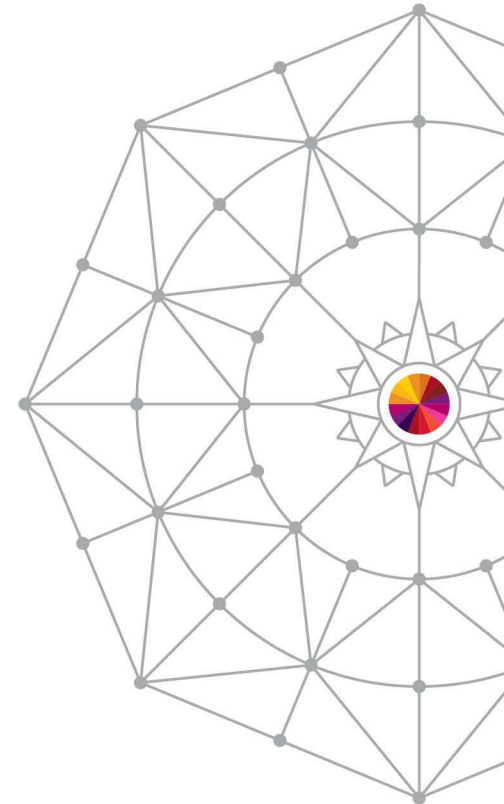
# Lab 10 – Test Your Routines

- Use ISMF 7.4 to test your routines
  - Option 1 for DEFINE new test
- Build a test case with the following rules:
  - Expected result: NULL
  - Description: Test1
  - DSN: SYS1.ANYTHING.ANYTHING
- Store it in SHARAx.S15097.ACS(TEST1)



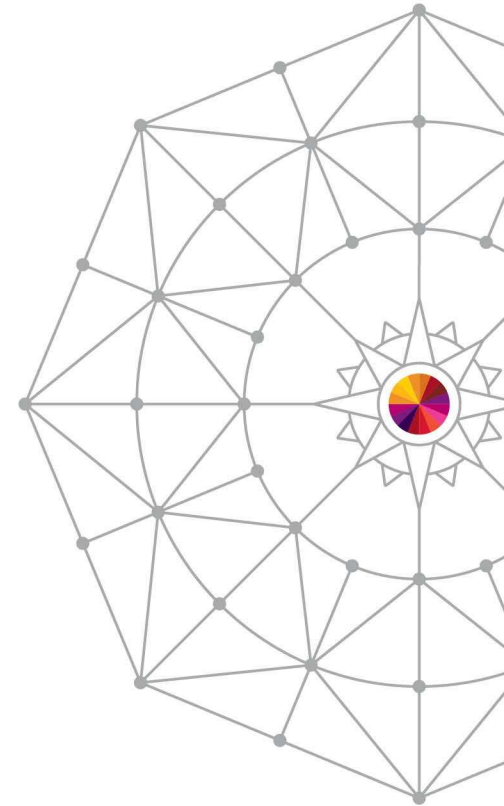
# Lab 10 – Test Your Routines

- Build another test case, TEST2, with rules:
  - Expected result: EXTENDED
  - Description: Extended Test
  - DSN: MY.EXT.DATASET
  - DSORG: VS
  - Size: 50000 (in KB)
  - Space\_Type: K
  - Second\_Qty: 50000
- Store it in SHARAx.x.S15097.ACS(TEST2)



# Lab 10 – Test Your Routines

- Run both tests (option 3)
  - CDS: SHARAx.x.S15097.SCDS
  - ACS Test Library: SHARAx.x.S15097.ACS
  - Select all routines
- What results do you get from each test?
- Do they match expectations?





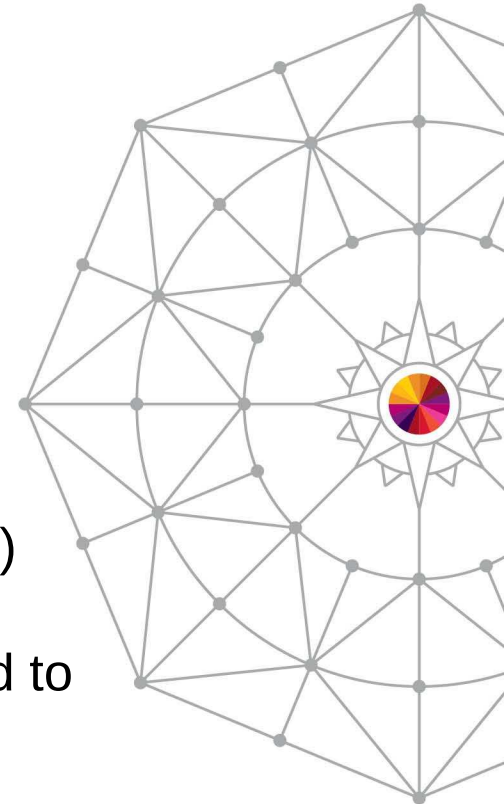
# Summary

- **Upon completion of this session, you should...**
  - Have a better understanding of the ACS environment
  - Be able to write a basic ACS routine
  - Understand how to Translate and Validate an ACS routine
  - Understand how to determine what Translate and/or Validate errors occurred
  - Be familiar with much of the ACS syntax
  - Be able to test your ACS routines



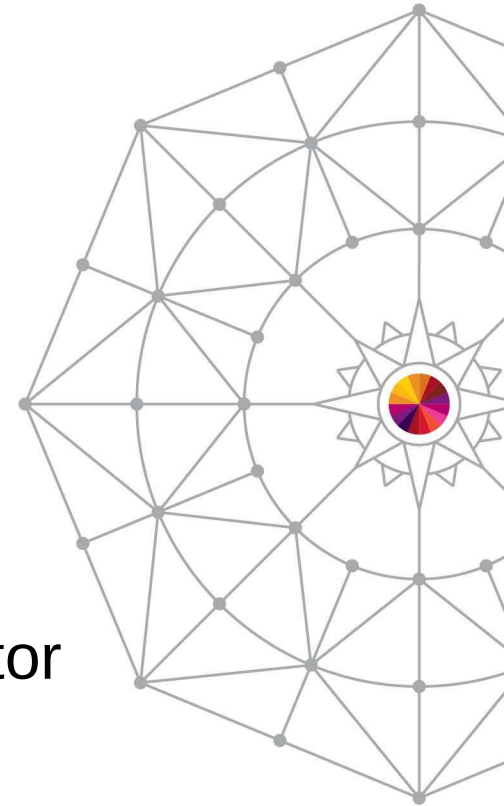
# Advanced Lab 11

- Create new DCA, SCA, MCA, SGA members in SHARAx.S15097.ACS
- Implement the following rules:
  - PDS and LIBRARY data sets are assigned to:
    - DC: PDSE, SC: Default, MC: none, SG: Default
  - Data sets over 100MB are assigned to:
    - DC: Extended, SC: Extended, MGMTCLAS: RLSIMM, SG: Extended
  - Data sets with HLQ of SYSTEM are assigned null (“”) for all routines
  - Data sets with second qualifier as “GS” are assigned to
    - SC: GSPACE
- Translate/validate/test your routines!



# For Additional Experimentation

- Create your own routines!
  - Available structures are on page 4 of the lab
- Create your own structures!
  - Just be sure to use SCDS of SHARAx.x.S15097.SMS.SCDS
- Try using the ACS Routine Testcase Generator
  - ISMF Option 11.1



# Notices & Disclaimers

Copyright © 2013 by International Business Machines Corporation.

No part of this document may be reproduced or transmitted in any form without written permission from IBM Corporation.

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 product(s) and/or programs(s) 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.

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.

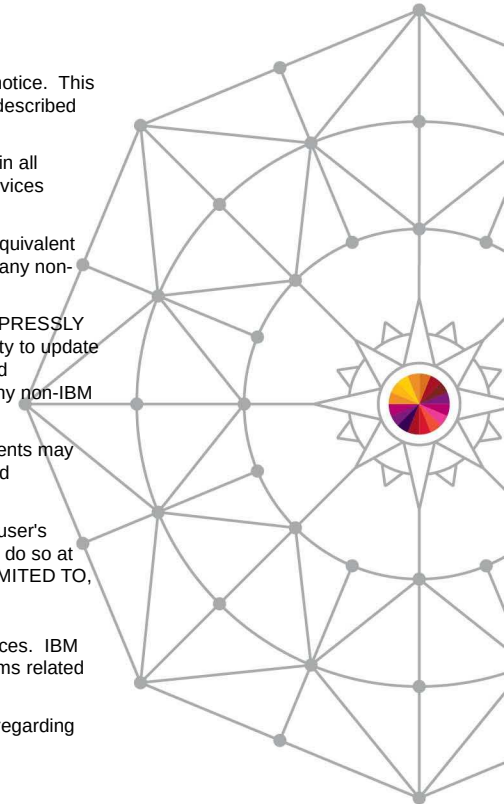
The responsibility for use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's or user's ability to evaluate and integrate them into their operating environment. Customers or users attempting to adapt these techniques to their 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 any other 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

DFSMSdftp, 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, Seascope, 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.

