IBM Education Assistance for z/OS V2R1

Item: New Function
Element/Component: DFSORT

Material is current as of June 2013

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

- IBM
- DFSORT
- Hiperspace
- z/OS

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.

Presentation Objectives

The purpose of this session is to provide a brief overview, usage and invocation information for each of the following new z/OS DFSORT V2R1 items:

- Functional Enhancements
  - Alphanumeric tests
  - Symbol enhancements
  - PARSE enhancements
  - Add string at end of VLR records
- Improve DFSORT/DB2 Synergy
  - Describe enhancements designed to improve how DFSORT interacts with DB2 Utilities
  - Describe how these enhancements benefit all users of DFSORT
- Dynamic Sort Enhancements
  - Describe enhancements for managing central storage usage
  - Review updates to DFSORT's installation defaults to control how DFSORT is to manage central storage usage

Presentation Objectives (continued)

The purpose of this session is to provide a brief overview, usage and invocation information for each of the following new z/OS DFSORT V2R1 items:

- Support for 64-Bit Callers with 64-Bit Addressed Records
  - Invoke DFSORT from 64-bit address mode programs
  - Pass 64-bit addressed records to DFSORT with exits
- RAS Enhancements
  - The following messages are updated to provide customers with better diagnostic information:
    - ICE118I
    - ICE015A
    - ICE083A
    - ICE141A
    - ICE285A

Overview: Functional Enhancements - Alphanumeric Tests

- Problem Statement / Need Addressed
  - Customers often want to do comparison tests to see if a field only contains characters in a specific set (e.g. A-Z, a-z and/or 0-9).

- Solution
  - Alphanumeric Tests: New UC, LC, MC, UN, LN and MN keywords (similar to the previously available NUM keyword) now allow you to test a field for various combinations of alphanumeric characters or non-alphanumeric characters using binary (BI) format.
    - UC: Uppercase characters (A-Z)
    - LC: Lowercase characters (a-z)
    - MC: Mixed case characters (A-Z, a-z)
    - UN: Uppercase and numeric characers (A-Z, 0-9)
    - LN: Lowercase and numeric characters (a-z, 0-9)
    - MN: Mixed case and numeric characters (A-Z, a-z, 0-9)

- Benefit / Value
  - This new support allows them to specify various sets of characters using a single compare condition. Previously, they would have had to code multiple compare conditions.

Usage & Invocation: Alphanumeric Tests

- Alphanumeric Tests Usage:
  - You can use alphanumeric test keywords(UC, LC, MC, UN, LN and MN) in the following comparison operands: COND, INCLUDE, OMIT, BEGIN, END, WHEN and TRLID.
  - Examples:
    - `INCLUDE COND=(11,10,BI,EQ,MC)`
    - `   OMIT COND=(50,5,BI,EQ,LC)`

- Parse Field Tests:
  - PARSE function can now be used with alphanumeric test keywords(UC, LC, MC, UN, LN and MN) to start or end when a character from any of various alphanumeric character sets.
  - Example:

```
INREC PARSE=(%01=(ENDBEFR=UC,FIXLEN=5)),BUILD=(%01)
```

Overview: Functional Enhancements - PARSE Enhancements

- Problem Statement / Need Addressed
  - Customers often have records with a very large number of delimited fields. Customers often have records with consecutive fields that they want to parse in the same way. Customers often want to parse fields based on a specific set of characters.

- Solution
  - PARSE Enhancements:
    - You can now use up to 1000 parsed fields (%0-%999) with the PARSE function; the previous limit was 100 parsed fields (%0-%99).
    - REPEAT=v is a new PARSE option that can be used to repeat a particular parse field definition multiple times.
    - STARTAFT=an, STARTAT=an, ENDBEFR=an and ENDAT=an can now be used with the PARSE function to start or end when a character from any of various alphanumeric character sets is found.

- Benefit / Value
  - This new support allows customers parse up to 1000 fields. The REPEAT=v allows them to easily ignore or process consecutive delimited fields of the same form. The alphanumeric tests allows them to specify various sets of characters using a single PARSE keyword.

Usage & Invocation: PARSE Enhancements

- More Parsed Fields:
  - You can now use up to 1000 parsed fields (%0-%999) with the PARSE function; the previous limit was 100 parsed fields (%0-%99).
  - Example:

```
OUTREC PARSE=(%121=(ENDBEFR=C',',FIXLEN=12),
              %322=(ENDBEFR=C',',FIXLEN=8),
              %999=(FIXLEN=5)),
       BUILD=(%121,X,%322,X,%999)
```

- Repeating Parse Fields:
  - REPEAT=v can be used with %n, %nn or %nnn to specify v identically defined consecutive parsed fields for which data is to be extracted. The parsed fields will start with the %n, %nn or %nnn field you select and be incremented by one for each repeated parsed field.
  - Example:

```
INREC PARSE=(%=(ENDBEFR=C',',REPEAT=3),
             %11=(ENDBEFR=C',',FIXLEN=10,REPEAT=4)),
      BUILD=(%11,X,%12,X,%13,X,%14)
```

Overview: Functional Enhancements - Symbol Enhancements

- Problem Statement / Need Addressed
  - Customers really like DFSORT symbols support and would like to see symbols supported for more DFSORT operands, especially those of the form KEYWORD=n.
- Solution
  - Symbol Enhancements:
    - KEYWORD=sym will be supported for operands of the form KEYWORD=n where n is a number.
- Benefit / Value
  - This new support allows them to use symbols for these types of keywords. Symbols can now be used with more DFSORT features such as ID=sym, SEQ=sym, ABSPOS=sym and FIXLEN=sym.

Usage & Invocation: Symbol Enhancements

- Symbol Enhancements:
  - KEYWORD=n can now be specified as KEYWORD=symbol where symbol represents an equivalent number (for example, if you have New_Length,25 in SYMNAMES, you can use LENGTH=New_Length wherever you can use LENGTH=25). The operands ABSPOS, ACCEPT, ADDPOS, AVGRLEN, DO, ENDPOS, ENDREC, FIXLEN, ID, IFOUTLEN, INCR, LENGTH, LINES, MAXLEN, RECORDS, REPEAT, SAMPLE, SEQ, SKIPREC, SPLIT1R, SPLITBY, START, STARTPOS, STARTREC, STOPAFT and SUBPOS can now have symbols.
  - Example:

```
OUTFIL REPEAT=Mult,
      IFOUTLEN=out_length,
      IFTHEN=(WHEN=GROUP,RECORDS=Num_records,
             PUSH=(id_col:ID=id_length))
```

Overview: Functional Enhancements - Add string at end of VLR record

- Problem Statement / Need Addressed
  - Customers often have variable-length files that need a particular string added to the end of each record.
- Solution
  - Add string at end of VLR records:
    - Add specific characters at the end of each VB record, such as X’0D0A’ (CRLF).
- Benefit / Value
  - This support allows them to add a string of 1-50 characters at the end of each VB record. Previously, they would have had to write their own E35 exit logic to add a string to each VB record.

Usage & Invocation: Add string at end of VLR record

- Add String at End of Variable-Length Records:
  - VLTRAIL=string is a new OUTFIL option that allows you to insert a character string (C'string') or hexadecimal string (X'yy...yy') at the end of each variable-length OUTFIL output record. You can add a string of 1-50 characters at the end of each VB record.
  - Example:

```
    OUTFIL VLTRIM=C' ',VLTRAIL=X'0D0A'
```

Migration & Coexistence Considerations: Add string at end of VLR record

- The following are new DFSORT/ICETOOL reserved words which are no longer allowed as symbols: LC, LN, MC, MN, UC and UN.

- If customers used any of these words as a symbol previously they must change them. For example, if they used MC, they can change it to mc.

Overview: RAS Enhancements

- Problem Statement / Need Addressed
  - Wanted to improve reliability and performance of DFSORT
    - Provide virtual storage constraint relief below 16MB
    - Reduce disk work space related failures
    - Improve scalability for very large sorts

- Solution
  - Exploit Extended TIOT, uncaptured UCB and above the line DSAB options for dynamically allocated work data sets
  - Expand “additional” work data sets capability provided in previous release
  - Increase maximum size of disk and memory object work files

- Benefit / Value
  - Improved reliability and scalability for all users of DFSORT

Usage & Invocation: Extended TIOT

- DFSORT V1R12 provided support for programs that invoke DFSORT, ICETOOL or ICEGENER and dynamically allocate input, output and work data sets using the options for Extended TIOT, uncaptured UCBs, and DSAB above 16 megabyte virtual.
  - These are options of dynamic allocation
- In DFSORT V2R1, these options are exploited by DFSORT's dynamic allocation of work data sets
  - Uncaptured UCBs (S99UCACB option) is always exploited
  - Extended TIOT (S99TIOEX option) and DSAB above 16 megabyte virtual (S99DSABA option) are used if DFOSRT is running authorized
- The exploitation is automatic.

Usage & Invocation: Additional work data sets

- DFSORT V1R12 provided capability for dynamic allocation of additional work data sets that are only used if needed
  - Primary space of zero
  - Secondary space only allocated if needed
- DFSORT V2R1 now provides capability to provide similar function for JCL or pre-allocated work data sets.
  - Work data sets with primary allocation of zero are only used when work data sets with non-zero primary have been exhausted
  - For example

```
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(10,5))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(10,5))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(0,5))
```

SORTWK03 is only used if no more space can be allocated to SORTWK01 and SORTWK02

Usage & Invocation: Increased size for work files

- To further exploit Extended Address Volumes, the maximum number of tracks that can be used for a single work data set has been increased from 1,048,576 to 16,777,216 when full track blocking is used
  - In cases where DFSORT is unable to use full track blocking, the limit may be less
- To further exploit large central storage configurations, the maximum amount of memory object storage that can be used as intermediate work space has been increased from 64 gigabytes to 1 terabyte

Overview: Dynamic Sort Enhancements

- Problem Statement / Need Addressed
  - Wanted better management over DFSORT's use of central storage
    - Over commitment of central storage and paging resources
    - Unbalanced use of resources used by concurrent sort applications
- Solution
  - Improved DFSORT installation defaults to control DFSORT's use of central storage
    - New TUNE installation default to favor storage or disk work space optimization
    - Storage related installation defaults can be specified as a percentage of current resources instead of a percentage of configured storage
  - Allocation of central storage in smaller increments
- Benefit / Value
  - Allows customers to tailor DFSORT's use of central storage usage based on system configuration and requirements
  - Gradual allocation in smaller increments allows for checking available resources and avoiding over commitment

Usage & Invocation: Dynamic Sort Enhancements

- New TUNE installation default
  - TUNE=STOR (shipped default)
    - DFSORT is to allocate storage in smaller increments
    - Increases allocation for dynamically allocated work data sets to reduce risk of failure if storage becomes constrained
  - TUNE=DISK
    - DFSORT allocates all storage to be used immediately
    - Reduces allocation for dynamically allocated work data sets based on storage already obtained
  - TUNE=OLD
    - DFSORT allocates storage as it did in prior release

- EXPOLD shipped default changed from MAX to 50%
  - Controls amount of storage in use by other applications (“old frames”) that DFSORT considers available for sorting
  - With TUNE=STOR, do not allow a sort to obtain more than 50% of old frames dynamically calculated at run time - If TUNE=OLD, then calculation is 50% of configured storage

- EXPRES shipped default changed from 0 to 10%
  - Controls amount of storage (available + old) that DFSORT considers unavailable for sorting applications
  - With TUNE=STOR, do not allow a sort to cause the reserved amount to fall below 10% of available frames calculated at run time
    - If TUNE=OLD, then calculation is 10% of configured storage

Migration & Coexistence Considerations: Dynamic Sort Enhancements

- Some sort applications may use less central storage in an effort to reduce impact to overall system performance
  - Reduced central storage usage may lead to an increase in disk work space usage and/or an increase in elapsed time
- To force DFSORT to operate as it did in z/OS V1R12/V1R13 set the following installation defaults:
  - TUNE=OLD
  - EXPOLD=MAX
  - EXPRES=0

Overview: Support for 64-Bit Callers with 64-Bit Addressed Records

- Problem Statement / Need Addressed
  - Increase use of “above bar” storage where “above bar” is defined as an address greater than a 2-gigabyte address.

- Solution
  - Eligible user programs and exits can now be written to:
    - Call DFSORT from a program in 64-bit address mode (AMODE 64)
    - Use DFSORT E15, E35 and E32 exits running in 64-bit address mode (AMODE 64)
    - Pass 64-bit addressed records to DFSORT using E15, E32, and E35 exits

- Benefit / Value
  - Less use of below bar storage and increased capacity of area to store records before exits pass records to DFSORT.

Usage & Invocation: Support for 64-Bit Callers with 64-Bit Addressed Records

- Call DFSORT from a program in 64-bit addressing mode (AMODE 64) using a new 64-bit invocation parameter list and the entry name ICEMAN64 or SORT64

- Use DFSORT E15, E35, and E32 exits running in 64-bit addressing mode (AMODE 64)

- Pass 64-bit addressed records to DFSORT using new 64-bit parameter lists for E15, E32, and E35 exits.

ICEPL64 DSECT

- The DFSORT target library, SICEUSER, contains a mapping macro named ICEPL64.

- Contains separate assembler DSECTs to be used for the invocation, E15, E32, and E35 64-bit parameter lists.

ICEPL64 DSECT Contents

```
*****************************************************************
* 64-Bit Invocation Parameter List for DFSORT
*****************************************************************
ICE64INV DSECT         64-Bit Invocation Parameter List
ICEPLID  DC    C'PL64SORT'     Identifier
ICEMDEX1 DS    C      Flag byte 1
ICE15A24 EQU   X'80'  AMODE 24 for E15 Exit or
ICE32A24 EQU   X'80'  AMODE 24 for E32 Exit
ICE15A31 EQU   X'40'  AMODE 31 for E15 Exit or
ICE32A31 EQU   X'40'  AMODE 31 for E32 Exit
ICE15A64 EQU   X'20'  AMODE 64 for E15 Exit or
ICE32A64 EQU   X'20'  AMODE 64 for E32 Exit
ICE35A24 EQU   X'10'  AMODE 24 for E35 Exit
ICE35A31 EQU   X'08'  AMODE 31 for E35 Exit
ICE35A64 EQU   X'04'  AMODE 64 for E35 Exit
ICE18A24 EQU   X'02'  AMODE 24 for E18 Exit
ICE18A31 EQU   X'01'  AMODE 31 for E18 Exit
ICEMDEX2 DS    C      Flag byte 2
ICERSVD1 EQU   X'80'  Reserved - must be set to zero
ICE39A24 EQU   X'40'  AMODE 24 for E39
ICE39A31 EQU   X'20'  AMODE 31 for E39
ICERSVD2 EQU   X'10'  Reserved - must be set to zero
```

ICEPL64 DSECT Contents (continued)

```
******************************************************************
* 64-Bit E15 Exit Parameter List for DFSORT
******************************************************************
ICE64E15 DSECT       64-Bit E15 Exit Parameter List
ICE15NR  DS    D      X'00000000' | Address of the new record
ICE15UC  DS    D      X'00000000' | User exit address constant
*
******************************************************************
* 64-bit E32 Exit Parameter List for DFSORT
******************************************************************
ICE64E32 DSECT       64-Bit E32 Exit Parameter List
ICE32NXF DS    D      X'00000000' | Increment of next file to be used
ICE32NIR DS    D      X'Address of next input record inserted from E32
ICE32UC  DS    D      X'00000000' | User exit address constant
*
******************************************************************
* 64-bit E35 Exit Parameter list for DFSORT
******************************************************************
ICE64E35 DSECT       64-Bit E35 Exit Parameter List
ICE35RL  DS    D      X'00000000' | Address of record leaving DFSORT
ICE35RO  DS    D      X'00000000' | Address of record in output area
ICE35UC  DS    D      X'00000000' | User exit address constant
```

64-bit Invocation Parameter List (Additional Detail)

- The invoking program uses the new 64-bit parameter list and must use ICEMAN64 or SORT64 as the entry point name for the LINK, ATTACH or XCTL macro. A 64--bit Register 1 points at the new 64-bit invocation parameter list described below:

Hex	Dec	Len	Bit	Contents
0	0	8		C’PL64SORT’
8	8	1	0	AMODE 24 E15 or E32
			1	AMODE 31 E15 or E32
			2	AMODE 64 E15 or E32
			3	AMODE 24 E35
			4	AMODE 31 E35
			5	AMODE 64 E35
			6	AMODE 24 E18
			7	AMODE 31 E18
9	9	1	0	Reserved. Must be set to 0.
			1	AMODE 24 E39
			2	AMODE 31 E39
			3	Reserved, Must be set to 0.
			4	E15 or E32 type of exit parameter list used and how Register 1 should be interpreted
			5	E35 type of exit parameter list used and how Register 1 should be interpreted
			6	E18 type of exit parameter list used and how Register 1 should be interpreted
			7	E39 type of exit parameter list used and how Register 1 should be interpreted
A	10	14		Reserved. Must be set to zeroes.
18	24	8		Address of control statement area (zeros if none)
20	32	8		X’00000000’ | Address of user exit E15 or E32 (zeros if none)
28	40	8		X’00000000’ | Address of user exit E35 (zeros if none)
30	48	8		X'00000000' | Address of user exit constant
38	56	8		Address of ALTSEQ Translation table (zeros if none)
40	64	8		X'00000000' | Address of ESTAE Area Pointer (zeros if none)
48	72	8		X'00000000' | Address of user exit E18 (zeros if none)
50	80	8		X'00000000' | Address of user exit E39 (zeros if none)
50	88	8		X’00000000’ | 4-Character call identifier (zeros if none)
60	96	40		Reserved. Must be set to zeros.

64-bit Invocation Parameter List (Additional Detail Continued)

- Reserved fields are defined. Must be set to zero as defined.
- When a 32-bit exit parameter list is used, a 32-bit Register 1 is returned from exit, and exit saves 32-bit registers in save area (Format-0 72 byte save area) pointed at by 32-bit Register 13.
- When a 32-bit Register 1 is returned, the high half of Register 1 will be ignored by DFSORT when the exit returns to DFSORT.
- When a 64 bit exit parameter list is used, a 64-bit Register 1 is returned from exit, and exit saves 64-bit registers in save area (Format-4 144 byte save area) pointed at by 64-bit Register 13.
- Address of exit must be a 'clean' 31-bit address or a 'clean' 24-bit address.
- Address of control statement area or ALTSEQ Translation table can be above or below the bar

64-bit E15 Exit parameter list

- DFSORT passes to the E15 exit a 64-bit general register 1 with an address of a parameter list that contains the record address and the user address constant. The format of the parameter list is:
  - Bytes 1 through 4 X'00000000' Address of the new record
  - Bytes 5 through 8 X'00000000' User exit address constant

- DFSORT provides a 144-byte Format 4 save area pointed to by 64-bit register 13 in which the exit can save the 64-bit registers.

- The E15 should expect and use 64-bit addresses.

- Before returning control to DFSORT, you must:
  - Place the address of the record in 64-bit register 1. This must be a 64-bit address, a clean 31-bit address or a clean 24-bit address.
  - Put the return code in 64-bit register 15.

64-bit E32 Exit Parameter List

- DFSORT passes to the E32 exit a 64-bit general register 1 with an address of a parameter list that contains the record address and the user address constant. The format of the parameter list is:
  - Bytes 1 through 8 X'00000000' Increment of next file to be used for input
  - Bytes 9 through 16 Address of next input record
  - Bytes 17 through 24 X'00000000' User exit address constant

- DFSORT provides a 144-byte Format 4 save area pointed to by 64-bit register 13 in which the exit can save the 64-bit registers.

- The E32 should expect and use 64-bit addresses.

- Before returning control to DFSORT, you must:
  - Place the address of the next input record from the requested input file in the second doubleword of the parameter list. This must be a 64-bit address, a clean 31-bit address or a clean 24-bit address.
  - Put the return code in 64-bit register 15.

64-bit E35 Exit Parameter List

- DFSORT passes to the E35 exit a 64-bit general register 1 with an address of a parameter list that contains the the 2 record addresses and the user address constant. The format of the parameter list is:
  - Bytes 1 through 8 X'00000000' Address of record leaving DFSORT
  - Bytes 9 through 16 X'00000000' Address of record in output area
  - Bytes 17 through 24 X'00000000' User exit address constant

- DFSORT provides a 144-byte Format 4 save area pointed to by 64-bit register 13 in which the exit can save the 64-bit registers.

- The E35 should expect and use 64-bit addresses.

- Before returning control to DFSORT, you must:
  - Place the address of the record in 64-bit register 1. This must be a 64-bit address, a clean 31-bit address or a clean 24-bit address.
  - Put the return code in 64-bit register 15.

MODS Control Statement New N64 Parameter

- When N64 is specified:
  - This indicates that your routine (exit) will use a 64-bit exit parameter list
  - It has already been bound or link-edited and can be used by DFSORT without further binding or link-editing

New Error Message ICE290A

ICE290A INVALID 64-BIT INVOCATION PARAMETER LIST - REASON CODE IS rsn

- DFSORT was invoked with a 64-bit invocation parameter list. However, an error was found in this parameter list. Reason code values (rsn) are indicated. DFSORT terminates.

- Reason code (rsn) values and programmer responses are listed in the DFSORT Messages and Codes publication

- **System Action:** DFSORT terminates

New Error Message ICE291I

ICE291I AMODE FLAG FOR EXIT WITHOUT CORRESPONDING EXIT ADDRESS IGNORED

- An AMODE flag was set for an exit for which a corresponding exit address was not present in the parameter list. For example, the AMODE 64 flag was set for the E15 exit, but the E15 exit address was zeros in the 64-bit invocation parameter list.

- **System action:** Processing continues. Each AMODE flag for an exit without a corresponding exit address is ignored.

- **Programmer response:** No action is necessary. If you want to eliminate this warning message, turn off the unneeded AMODE flag or specify a corresponding exit address.

Updated Error Message ICE034A

ICE034A MODS STATEMENT OPERAND ERROR

- New reason code was added for the case when MODS N64 parameter is used incorrectly.
  - N64 was specified for the fourth parameter, but DFSORT was not invoked using a 64-bit invocation parameter list.
- **System action:** The program terminates.
- **Programmer response:** N64 is not specified as the fourth parameter if DFSORT was not invoked using a 64-bit invocation parameter list.

Location of Example

- Example of Sort with 64-bit Parameter Lists can be found in:

Publication: z/OS DFSORT Application Programming Guide
Chapter: Examples of DFSORT Job Streams
Section: Sort Examples
Example: Sort with 64-bit Parameter Lists, E15, E35 and OUTFIL

Overview: RAS Enhancements (ICE118I)

- Problem Statement / Need Addressed
  - The message text for ICE118I was not worded strongly enough.

- Solution
  - Updated the ICE118I text so it is clear that we expect customers to specify a file size when DFSORT can not determine one.

- Benefit / Value
  - DFSORT requires an accurate file size estimate to determine the amount of storage required for the job to complete. Providing DFSORT with accurate file sizes will help ensure that we allocate the correct amount of storage.

Overview: RAS Enhancements (ICE118I)

- Original Message Text:
  - ICE118I:
    - UNKNOWN FILE SIZE – FILSZ=EN MAY IMPROVE RESOURCE USAGE AND PERFORMANCE

- Updated Message Text:
  - ICE118I:
    - UNKNOWN FILE SIZE - RECOMMEND SPECIFYING FILSZ=EN TO REDUCE RISK OF FAILURE OR DEGRADED PERFORMANCE

Overview: RAS Enhancements (ICE015A)

- Problem Statement / Need Addressed
  - The original ICE015A message explanation was too generic to pin point the actual error that was causing the abend

- Solution
  - Update the message to list the specific reasons as to why the message was issued.

- Benefit / Value
  - The updated explanation of the message will assist customers in understanding the reason that they received the ICE015A.

Overview: RAS Enhancements (ICE015A)

- ICE015A VARIABLE RECORD TOO SHORT

Explanation: Critical. This message was issued for one

of the following reasons:

- A record passed to DFSORT from user exit E15 or E35 contained a length in an RDW less than 4.
- An input data set contained a length in an RDW less than 4.
- An input data set contained a length in a BDW less than 8.
- A variable-length record was too short to contain all fields or, if Blockset was not selected, shorter than L4.
- A record did not contain the first byte of the first control field when VLSHRT was in effect and Blockset was not selected.

Overview: RAS Enhancements (ICE083A)

- Problem Statement / Need Addressed
  - Message ICE083A is issued when dynamic allocation for work data sets fails. DFSORT either returns the SVC 99 error (S99ERROR) or a hard coded value that we set.
- Solution
  - This enhancement will update the ICE083A message to list the device error along with an S99ERSN SMS reason code if available.
  - This item will also update the hard coded values to make it clear that they were set by DFSORT and not returned in S99ERROR by dynamic allocation.
- Benefit / Value
  - The updated reason codes will make it easier for customers to resolve the ICE083A error on their own.

Overview: RAS Enhancements (ICE083A)

ICE083A RESOURCES WERE UNAVAILABLE FOR DYNAMIC ALLOCATION OF WORK DATA SETS (derr) (srsn)

Explanation:

Critical. Dynamic allocation of sort work data sets could not be attempted or failed. derr indicates the error code associated with the dynamic allocation failure as set by DFSORT, or as returned in field S99ERROR by the system DYNALLOC service used by DFSORT.

One of the following derr values is set by DFSORT when dynamic allocation could not be attempted:

- I64K: The work space needed for each work data set exceeded 64K tracks for a single, physical data set, but DFSORT was unable to allocate the work data sets as large format.
- I999: The work space needed for each work data set exceeded an internal DFSORT limit.
- IDEV: Conflicting device types were received for the work data sets.

If the derr value does not start with I, DFSORT attempted to dynamically allocate work data sets using the DYNALOC/DYNALLOC and DYNSPC options in effect but the attempt failed and the derr value displays the return code returned by the dynamic allocation.

Overview: RAS Enhancements (ICE083A)

Programmer Response:

- If derr is I64K: Use the DYNALLOC=(,n) parameter to increase the maximum number (n) of work data sets such that the work space needed for each work data set does not exceed 64K tracks. Alternatively, if the VIO=YES installation option is in effect, setting the VIO=NO installation option may allow DFSORT to allocate the work data sets as large format.

- If derr is I999: Use the DYNALLOC=(,n) parameter to increase the maximum number (n) of work data sets; this will reduce the work space needed for each work data set.

- If derr is IDEV: Ensure that each work data set has the same device type.

- If derr is 97xx: An SMS failure occurred. 97xx is the S99ERROR code (in hexadecimal) associated with the failure. If srsn is displayed in the message, it is the S99ERSN SMS reason code (in hexadecimal) associated with the failure. See z/OS MVS Programming Authorized Assembler Services Guide for an explanation of the S99ERROR code (derr) and the S99ERSN SMS reason code (srsn). Take the appropriate indicated action to correct the error.

- If derr is none of the above: derr is the S99ERROR code (in hexadecimal) associated with the failure. See z/OS MVS Programming Authorized Assembler Services Guide for an explanation of the S99ERROR code (derr). Take the appropriate indicated action to correct the error.

Overview: RAS Enhancements (ICE141A)

- Problem Statement / Need Addressed
  - Message ICE141A can be issued for 4 reasons. The message explanation listed all 4 reasons as possibilities, but it did not indicate which of the 4 reasons had occurred.
- Solution
  - Update message ICE141A to include reason codes.
  - Also, update the PEERVALE path of code to issue ICE204A instead of the ICE141A.
- Benefit / Value
  - Providing reason codes in the ICE141A will allow customers who receive this message to pinpoint the specific reason for the message and the appropriate action to take.
  - Updating the PEERVALE path to issue ICE204A, which will more accurately reflect the error being detected.

Overview: RAS Enhancements (ICE141A)

ICE141A SPANNED RECORD ON ddname COULD NOT BE ASSEMBLED - REASON CODE IS rsn

Explanation:

Critical. A spanned record on the indicated data set could not be properly assembled.

rsn is the reason code associated with the error. The reason code helps you determine the specific cause of the error. Reason code values (rsn) are as follows:

1. A segment length was greater than the LRECL.
2. A segment length was less than 4 bytes.
3. The total length of segments was greater than the LRECL.
4. Segments are blocked incorrectly for VBS data sets (that is a first and a last segment are in the same block).

Overview: RAS Enhancements (ICE141A)

Programmer Response:

Take the action for the indicated reason code value (rsn) as follows:

1. Ensure all records in the indicated data set have segment lengths that are less than the LRECL.
2. Ensure all records in the indicated data set have segment lengths that are at least 4 bytes.
3. Ensure all records in the indicated data set have segment lengths that total less than the LRECL.
4. Ensure all records in the indicated VBS data set are blocked correctly.

Overview: RAS Enhancements (ICE141A)

PEERVALE Path Changes:

DFSORT will now issue ICE204A instead of ICE141A.

ICE204A INCOMPLETE SPANNED RECORD DETECTED ON ddname - RC=16

Overview: RAS Enhancements (ICE285A)

- Problem Statement / Need Addressed
  - The message text for ICE285A was not clear.

- Solution
  - Updated the ICE285A text so it is clear that DFSORT requires BLOCKSET for sort work data sets that are greater than 17476 cylinders.

- Benefit / Value
  - Certain functions are only available on the BLOCKSET path. This update is to make it clear that BLOCKSET is required for this situation.

Overview: RAS Enhancements (ICE118I)

- Original Message Text:
  - ICE285A:
    - BLOCKSET IS REQUIRED FOR WORK DATA SETS ON VOLUMES WITH MORE THAN 17476 CYLINDERS

- Updated Message Text:
  - ICE285A:
    - DEVICE WITH CAPACITY OVER 17476 CYLINDERS REQUIRES BLOCKSET FOR ALLOCATED WORK DATA SETS

Migration & Coexistence Considerations

- The format of the following messages has been changed. If you have automation software monitoring any of these messages, then you need to update the automation software.
  - ICE118I
  - ICE083A
  - ICE141A
    - If your automation software is monitoring this message you may want to add ICE204A and ICE197I to the automation software.
  - ICE285A

Presentation Summary

We have covered each of the following new z/OS DFSORT V2R1 items:

- Functional Enhancements
  - Alphanumeric tests
  - Symbol enhancements
  - PARSE enhancements
  - Add string at end of VLR records
- Improve DFSORT/DB2 Synergy
  - Describe enhancements designed to improve how DFSORT interacts with DB2 Utilities
  - Describe how these enhancements benefit all users of DFSORT
- Dynamic Sort Enhancements
  - Describe enhancements for managing central storage usage
  - Review updates to DFSORT's installation defaults to control how DFSORT is to manage central storage usage

Presentation Summary (continued)

We have covered each of the following new z/OS DFSORT V2R1 items:

- Support for 64-Bit Callers with 64-Bit Addressed Records
  - Invoke DFSORT from 64-bit address mode programs
  - Pass 64-bit addressed records to DFSORT with exits

- RAS Enhancements
  - The following messages are updated to provide customers with better diagnostic information:
    - ICE118I
    - ICE015A
    - ICE083A
    - ICE141A
    - ICE285A

Appendix

- Publications
  - z/OS DFSORT Application Programming Guide (SC23-6878-00)
  - z/OS DFSORT Messages and Codes (SC23-6879-00)
  - z/OS DFSORT: Getting Started (SC23-6880-00)
  - z/OS DFSORT Installation and Customization (SC23-6881-00)
  - z/OS DFSORT Tuning Guide (SC23-6882-00)

- Web site: http://www.ibm.com/storage/dfsort

- Contacts:
  - DFSORT Hotline - dfsort@us.ibm.com