



Data Extraction Programs User's Guide for Conversion to DFSMSrmm

Version 2 Release 1

First Edition (June 2011)

This edition applies to z/OS DFSMSrmm V2R1, and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

Comments may be addressed to:

International Business Machines Corporation
Department 61D, Building 031-2
9000 S. Rita Road
TUCSON, AZ 85744
U.S.A

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright IBM Corporation 2012.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Tables	vii	Return Codes.	58
Figures	ix	Messages	59
Notices	xi	Creating VRS Records using EDGCSVDS	60
About This Book	xiii	Sample JCL	61
Trademarks	xv	Execution Parameter	61
Summary of changes	xvii	Input Data Sets	62
Chapter 1. Before You Begin	1	Output Data Sets	62
Chapter 2. Introduction	7	Return Codes.	62
Chapter 3. Conversion from TMS/CA-1	9	Messages	63
Preparing to run the Extract	10	Policy Conversion Considerations	64
Conversion of TMS/CA-1 Pool Definitions and Rules	13	RDS and VPDD Input Requirements	64
Input	13	GDG vs. Non-GDG.	65
EDGRCSCR - Process CA-1 Pool Definitions	14	Policy Conversion Differences	65
Extracting Data from the TMC Records using EDGC5LDR	24	Hints and Tips	66
Sample JCL	25	Chapter 4. Conversion from TLMS	69
Execution Parameters	27	Functional Description.	70
Required Data	29	Preparing to run the Extract Program.	70
Input Data Sets	31	The EDGCDYNM Program	70
Output Data Sets	34	Input Files.	71
Return codes	35	Output Files	71
Messages	36	Input Parameters	71
Exported Logical Volumes and Stacked Volumes Support	43	Program Execution	77
Creating the Empty Bin Records using EDGC5BIN	44	Program Limitations	77
Planning	44	Return Codes.	78
Sample JCL	45	Messages	78
Execution Parameter	45	TLMS Conversion Considerations	83
Input Data Sets	46	Storage Locations	83
Output Data Sets	47	Shelf Management	84
Return Codes.	48	Retention Criteria	85
Messages	48	Scratch Volumes	87
Overview of Policy Conversion.	50	Exported Logical Volumes & Stacked Volumes.	88
Creating VRS Records using EDGCSRDS	51	Parallel Running.	88
How the Conversion Process Works	52	Chapter 5. Conversion from Manual Management to DFSMSrmm	89
Sample JCL	53	Sample JCL	89
Execution Parameters	55	EDGJVTC1	89
Input Data Sets	56	EDGJVTC2	89
Output Data Sets	56	EDGJVTC3	90
Return Codes.	56	Sample REXX Procedure	90
Messages	56	EDGCVTC0	90
Creating SMS ACS Statements using EDGRACS	57	EDGCVTC4	90
Input Data Sets	58	EDGCVTC5	91
Output Data Sets	58	EDGCVTC6	92
		Chapter 6. Conversion from TAPE2000	93
		Functional Description.	94
		Preparing to run the Extract Program.	94
		The EDGRT2EX Exec	95
		Input Files.	96
		Output Files	96
		Converted Data Fields.	97
		Execution	97
		Return Codes	100

Messages	100
Summary Messages	111
TAPE2000 Conversion Considerations	111
Storage Locations	111
Shelf Management	112
Retention Criteria	112
Parallel Running	113

Chapter 7. Conversion from Control-M/Tape 115

Data Extraction	116
Control-M/Tape MDB records	116
Create an extended Control-M/Tape record (EDGRCTT1)	116
Extracting data from the extended MDB records (EDGRCTT2)	120
Extracting data from the RULES file (EDGRCTT3)	130
Conversion Considerations	135
Volume Ownership	135
Tape Pool Management	135
Storage Location Management	136
Retention Management	137
Parallel Processing	138

Chapter 8. Conversion from ASG-ZARA/AUTOMEDIA 141

Unload the ASG-ZARA/AUTOMEDIA DB Records (ZARAXTRT)	142
Create Extract Data without HEADER and TRAILER Records (EDGRCZA1)	142
Input Data Sets	145
Output Data Sets	145
Return Codes	146
Messages	146
Convert Data out of the ASG-ZARA/AUTOMEDIA sorted Records (EDGRCZA2)	148
Input Data Sets	150
Output Data Sets	150
Return Codes	151
Messages	151
Create VRS records from the ASG-ZARA/AUTOMEDIA Expiration Report (EDGRCZA3)	168
Input Data Sets	169
Output Data Sets	170
Return Codes	170
Data set name mask and jobname mask correcting	170
Messages	171
Am i ready to continue?.	182

Chapter 9. Building the DFSMSrmm CDS 185

Preparing to run EDGCNVT	185
System-managed Volumes	185
Pool Prefixes	186
VRS Conversion	186
Movement Tracking Date	187
EDGCNVT Considerations for Expiration Date	187
Running EDGCNVT and EDGCEBIN	187

Execution Parameters	189
EDGCNVT Input Files	190
EDGCNVT Output Files	202
EDGCNVT Return Codes	204
EDGCNVT User Exit	204
EDGCNVT Messages	205
EDGCEBIN Input Files	215
EDGCEBIN Output Files	215
EDGCEBIN Return Codes	216
EDGCEBIN Messages	216
Creating the DFSMSrmm CDS	217
Run EDGJLOAD	217
Create DFSMSRmm CDS Control Record	219
Verifying the DFSMSrmm CDS	220
Running DFSMSrmm Commands	221
Adding Empty Bins	221
Adding VRSs	221

Chapter 10. Running in parallel with RMM 223

Exit Modifications	223
Exit Overview	223
Activation of Parallel Running Exits	223

Chapter 11. EDGCVRSX (sample of EDGUX100) 227

Overview	227
Introduction	227
Functions	227
Prerequisites to use EDGCVRSX	228
How to reload the dynamic table	229
Rules how EDGCVRSX finds a match in dynamic table	229
Messages	231
EDGCVRSX (Macro)	232
Function	232
Invocation	232
Parameter description	232
EDGCVRSX (UXTABLE loader)	233
Function	233
Invocation	233
Messages	233
EDGCVRSX (Procedure to start EDGCVRSX)	234
Function	234

Appendix A. Modification Required for EPIC/MVS and RMM Parallel Run . . . 235

Appendix B. User Survey 245

Conversion to RMM Process Customer Critique	245
Overall Satisfaction Rating	245
Data Extraction Phase Assessment	246
Data Extraction	246
Delivery	246
Extracted Data Conversion Phase Assessment	246
Data Conversion	246
Delivery	247
Post-processing (CDS Creation) Phase Assessment	247
CDS Creation	247

Delivery 247
Quality and Value 248

Data Extraction 248
Data Conversion 248

Tables

1.	Summary of changes	xvii	46.	DFSMSrmm-supplied conversion samples, programs, and JCL	93
2.	DFSMSrmm-supplied conversion samples	3	47.	EDGRT2EX Output Data Sets and Records	94
3.	EDGRCSR Input Data Sets	15	48.	EDGRT2EX Input Files	96
4.	EDGRCSR Output Data Sets	16	49.	EDGRT2EX Output Files	96
5.	EDGRCSR Return Codes	18	50.	EDGRT2EX Return Codes	100
6.	EDGC5LDR Modifiable Fields	29	51.	DFSMSrmm supplied conversion samples, programs, and JCL	115
7.	EDGC5LDR Input Data Sets	31	52.	EDGRCTT1 Input Data Sets	118
8.	EDMNAME Identification Data Set Record Format	31	53.	EDGRCTT1 Output Data Sets	118
9.	VRSMGMT Management Value Data Set Record Format	32	54.	EDGRCTT1 Return Codes	119
10.	STORE Location Name Data Set Record Format	33	55.	EDGRCTT2 Input Data Sets	120
11.	LOAN Location Name Data Set Record Format	34	56.	EDGRCTT2 Output Data Sets	122
12.	SGNAME Pool Table Record Format	34	57.	EDGRCTT2 Return Codes	122
13.	EDGC5LDR Output Data Sets	35	58.	EDGRCTT3 Input Data Sets	130
14.	EDGC5LDR Return Codes	35	59.	EDGRCTT3 Output Data Sets	131
15.	EDGC5BIN Input Data Sets	46	60.	EDGRCTT3 Return Codes	131
16.	STORE DD Input File Field Description	46	61.	Control-M/Tape retention methods and DFSMSrmm VRS equivalences	138
17.	EDGC5BIN Output Data Sets	47	62.	DFSMSrmm-supplied samples for conversion from ASG-ZARA/AutoMedia	141
18.	EDGC5BIN Return Codes	48	63.	EDGRCA1 Input Data Sets	145
19.	EDGCSRDS Input Data Sets	56	64.	EDGRCA1 Output Data Sets	145
20.	EDGCSRDS Output Data Sets	56	65.	EDGRCA1 special DETAIL DD	145
21.	EDGCSRDS Return Codes	56	66.	EDGRCA1 Return Codes	146
22.	EDGRACS Input Data Sets	58	67.	EDGRCA2 Input Data Sets	150
23.	EDGRACS Output Data Sets	58	68.	EDGRCA2 Output Data Sets	150
24.	EDGRACS Return Codes	58	69.	EDGRCA2 special DETAIL DD	151
25.	EDGCSVDS Input Data Sets	62	70.	EDGRCA2 Return Codes	151
26.	EDGCSVDS Output Data Sets	62	71.	EDGRCA3 Input Data Sets	169
27.	EDGCSVDS Return Codes	62	72.	EDGRCA3 Output Data Sets	170
28.	Character conversion samples	64	73.	EDGRCA3 special DETAIL DD	170
29.	DFSMSrmm supplied conversion samples, programs, and JCL	69	74.	EDGRCA3 Return Codes	170
30.	EDGCDYNM Output Data Sets and Records	70	75.	EDGCNVT Execution Parameters	189
31.	EDGCDYNM Input Files	71	76.	EDGCEBIN Execution Parameters	190
32.	EDGCDYNM Output Files	71	77.	EDGCNVT Input Files	190
33.	DFSMSrmm-supplied conversion samples, programs, and JCL	89	78.	EDGCNVT Output Files	202
34.	EDGCVTC0 Input Data Sets	90	79.	EDGCNVT Return Codes	204
35.	EDGCVTC0 Output Data Sets	90	80.	EDGCEBIN Input Files	215
36.	EDGCVTC0 Return Codes	90	81.	EDGCEBIN Output Files	215
37.	EDGCVTC4 Input Data Sets	90	82.	EDGCEBIN Return Codes	216
38.	EDGCVTC4 Output Data Sets	91	83.	Dynamic table sample (UXTABLE)	230
39.	EDGCVTC4 Return Codes	91	84.	Sample JCL parameter & values assigned from EDGCVR SX based on above table	231
40.	EDGCVTC5 Input Data Sets	91	85.	Data Extraction Phase Assessment	246
41.	EDGCVTC5 Output Data Sets	91	86.	Extracted Data Conversion Phase Assessment	247
42.	EDGCVTC5 Return Codes	92	87.	Post-processing (CDS Creation) Phase Assessment	247
43.	EDGCVTC6 Input Data Sets	92	88.	Quality and Value Extraction Assessment	248
44.	EDGCVTC6 Output Data Sets	92	89.	Quality and Value Conversion Assessment	248
45.	EDGCVTC6 Return Codes	92			

Figures

1. DFSORT: Sample JCL	11	37. (Part 3 of 5). Sample JCL to Execute EDGRT2EX	98
2. EDGC5LDR: Sample JCL	15	38. (Part 4 of 5). Sample JCL to Execute EDGRT2EX	99
3. EDGC5LDR: Sample JCL, CA-1 Release 5.0 and above	25	39. (Part 5 of 5). Sample JCL to Execute EDGRT2EX	100
4. EDGC5LDR Execution Parameter Syntax	27	40. EDGRT2EX Run Summary Messages example	111
5. EDGC5LDR Execution Parameters: Sample 1	28	41. Unload Control-M/Tape MDB using IOADUL utility	116
6. EDGC5LDR Execution Parameters: Sample 2	29	42. Sample JCL to Execute EDGRCTT1	117
7. EDGC5LDR Execution Parameters: Sample 3	29	43. Sample JCL to Execute EDGRCTT2	120
8. EDGCNVT Storage Location Name Reassignment	29	44. Sample JCL to Execute EDGRCTT3	130
9. EDGCNVT SYSIN Control Statement Format for VMEDIA	30	45. Unload ASG-ZARA/AutoMedia DB using ZARAXTRT utility	142
10. EDGCNVT SYSIN Control Statement Format for OWNER	30	46. Sample JCL to Execute EDGRCZA1	143
11. EDGC5LDR: EDM Identification Sample	32	47. Sample JCL to Execute EDGRCZA2	149
12. EDGC5LDR: VRS Management Value Name Sample	32	48. Sample JCL to Execute EDGRCZA3	169
13. EDGC5LDR: STORE DDNAME Sample	33	49. Sample EDGCNVT and EDGCEBIN Execution JCL	188
14. EDGC5LDR: LOAN DDNAME Sample	34	50. Convert Program VRS Input Record Format	191
15. EDGC5BIN: Sample JCL, CA-1 Release 5.0 and above	45	51. EDGCNVT Record Create Date, Time, CDS and System Identification	191
16. EDGC5BIN: STORE DDNAME Sample	47	52. VRSREL Release action	192
17. Example of Data Produced from the RDS	52	53. RETACTN Release action	192
18. The VRS built by EDGCSRDS	53	54. IF STORLOC SYSIN Statement.	193
19. EDGCSRDS: Sample JCL	54	55. IF STORLOC Example with LOCDEF	193
20. EDGCSVDS: Sample JCL	61	56. IF STORLOC Onsite Example	193
21. EDGCDYNM SYSIN: OPTIONS Control Statement	72	57. IF STORLOC Offsite Example	193
22. EDGCDYNM SYSIN: VRSNAMEP Control Statement	73	58. IF STORLOC Remote Example.	193
23. EDGCDYNM SYSIN: VRSMGMTP Control Statement	73	59. IF STORLOC with Blank Example	194
24. EDGCDYNM SYSIN: TLMSLOCN Control Statement	74	60. IF DSN SYSIN Statement Syntax	194
25. EDGCDYNM SYSIN: DFRMMLOC Control Statement	74	61. IF VOL SYSIN Statement Syntax	194
26. EDGCDYNM SYSIN: VOLMEDIA Control Statement	75	62. IF OWNER SYSIN Statement Syntax for Default Owner	194
27. EDGCDYNM SYSIN: SKIPVOL Control Statement	76	63. IF OWNER SYSIN Statement Syntax for Translating Owner	195
28. EDGCDYNM SYSIN: DEFOWNER Control Statement	76	64. LOCDEF Statement	195
29. EDGCDYNM SYSIN: VRSOWNER Control Statement	77	65. VOLPRE SYSIN Statement	196
30. EDGCDYNM Sample JCL.	77	66. VOLPRE Example	197
31. Sample Retention Selection Report for Conversion (left side)	85	67. VOLRANGE with LOCATION Statement	197
32. Sample Retention Selection Report for Conversion (right side).	86	68. VOLRANGE with PREFIX Statement	198
33. Sample Converted VRS Definitions.	86	69. VOLRANGE with MEDIANAME statement	198
34. Sample JCL to Execute TPMRPEXT	95	70. VOLRANGE with SGNAME statement	198
35. (Part 1 of 5). Sample JCL to Execute EDGRT2EX	97	71. VOLRANGE with TYPE statement	199
36. (Part 2 of 5). Sample JCL to Execute EDGRT2EX	97	72. VOLRANGE with NORACK statement	199
		73. VMEDIA Statement with MEDIANAME	200
		74. VMEDIA with MEDIANAME Statement	200
		75. OPTION RM.	200
		76. OPTION EXPDT_RETAINBY	201
		77. OPTION EXPDT_LASTREF	202
		78. Example of SYSOUT Contents	203
		79. Example of VRSCMDS Contents	203
		80. (Part 1 of 2). Sample EDGJLOAD Execution JCL	218
		81. (Part 2 of 2). Sample EDGJLOAD Execution JCL	219

82.	Sample EDGUTIL Execution JCL for VERIFY Processing	220	85.	Code to Modify for Parallel Exit Run	225
83.	Sample USERMOD	224	86.	JCL to Modify RMM for EPIC/MVS Parallel Run.	235
84.	Sample JCL to Link the IGXMSGEX Exit	225			

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the responsibility of the user.

For online versions of this book, we authorize you to:

1. Copy, modify, and print the documentation contained on the media, for use within your enterprise, provided you reproduce the copyright notice, all warning statements, and other required statements on each copy or partial copy; and
2. Transfer the original unaltered copy of the documentation when you transfer the related IBM product (which may be either machines you own, or programs, if the program's license terms permit a transfer). You must, at the same time, destroy all other copies of the documentation.

You are responsible for payment of any taxes, including personal property taxes, resulting from this authorization.

THERE ARE NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

Your failure to comply with the terms above terminates this authorization. Upon termination, you must destroy your machine readable documentation.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, 208 Harbor Drive, Stamford, CT 06904, U.S.A.

About This Book

This publication is intended to help the customer to convert to DFSMSrmm from other tape management products like:

- TMS/CA-1
- TLMS
- Manual Management
- TAPE2000
- CONTROL-M/TAPE
- ASG-ZARA/AUTOMEDIA

Trademarks

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

IBM
DFSMSrmm
DFSMSdss
DFSMSdfp
DFSMShsm
DFSORT
MVS/ESA
DFSMS
DFSMS/MVS
MVS/DFP
RACF

Other trademarks:

CA-1 is a registered trademark of CA Technologies
CA-Dynam is a registered trademark of CA Technologies
ASG-Zara is a registered trademark of ASG Software Solutions
Tape2000 is a trademark of SEA Software Engineering of America

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

Summary of changes

Table 1. Summary of changes

APAR / Development	Description
OA38101	The following changes for EDGRCT2 module: <ul style="list-style-type: none"> • Insure that LVFLAGD is not overlaid by new values during EDGRCT2 program processing
z/OS V2R1	The following global changes were done for all parts that belong to the extraction phase: <ul style="list-style-type: none"> • Remove front end code from SYS1.SAMPLIB. • Make removed parts available as web download, on the Z/OS web site designated for “as-is” downloads (http://www.ibm.com/systems/z/os/zos/downloads/#asis). • Update conversion documentation, previously shipped in SYS1.SAMPLIB as EDGMM01, to a workstation-readable PDF file (this document). • Code that is common to all product conversions (“non front-end”) code, namely EDGCNVT, will be shipped as previously.
z/OS V2R1	The following changes were done in EDGCNVT: <ul style="list-style-type: none"> • The OPTION EXPDT_RETAINBY retainby_value statement has been added to the list of EDGCNVT SYSIN statements. • The OPTION EXPDT_LASTREF extra_days statement has been added to the list of EDGCNVT SYSIN statements. • Support for selection of the volume attribute EXPDT_RETAINBY at volume level is provided via LVEXPDT_RETAINBY field in 'L record' EDGCLREC. • Support for selection of the data set attribute EXPDT_LASTREF extra days on data set level is provided via LDS1LRED field in 'L record' EDGCLREC and via DSNLRED flag in 'D record' EDGCDREC. • “EDGCNVT Considerations for Expiration Date” on page 187 has been added to describe the processing of volume and data set expiration dates at conversion time.
OA37254, OA37482 (FIN APARs, changes are included in V2R1)	The following changes were done: <ul style="list-style-type: none"> • The descriptions of some EDGCDYNM messages were added. • The possibility to use SKIPVOL VOLPFX sysin statement for EDGCDYNM program was deleted. • The EXPIRYDATEIGNORE flag is set for the specific expiration date 99000 during running EDGCDYNM.
z/OS V1R13	The following changes were done to installation exit support: <ul style="list-style-type: none"> • New keyword RM in EDGCVRSR to supply retention method. • New keyword VX in EDGCVRSR to set VRSELEXCLUDE attribute. <p>The following changes were done in EDGCNVT conversion support:</p> <ul style="list-style-type: none"> • Add the OPTION RM retention method statement to the list of EDGCNVT SYSIN statements. • Support for selection of the retention method on volume level is provided via LVRMVR and LVRMEXP flags in 'L record' EDGCLREC. • Support for excluding data sets from VRSEL processing on data set level is provided via LDS1VRSX flag in 'L record' EDGCLREC and via DSVRSX flag in 'D record' EDGCDREC. • Regardless of retention method, the expiration date of the first data set record of a multi volume data set is propagated to all other data set records of the multi volume data set.
OA34429	The following changes are for EDGCNVT: <ul style="list-style-type: none"> • Using the value of sequence number of data set one from EDGCLREC (volume data) corrects the logical sequence number for the first data set in a multi-volume/multi-datasets chain for a scratch volume.

Table 1. Summary of changes (continued)

APAR / Development	Description
OA34078	The following changes are for CA-1 conversion: <ul style="list-style-type: none"> • The higher location priority of stacked volumes in a STORE location is honored. • No replacement of location data is performed when you run EDGUTIL with MEND(VOLUME) to enable stacked volume support.
OA33551	The following changes are for EDGC5LDR: <ul style="list-style-type: none"> • Using 64-bit arithmetic instead of 32-bit for correct calculation the size of the data set in K-bytes.
OA32770	The following changes are for CA-1 conversion: <ul style="list-style-type: none"> • IBM system managed library information is used to determine if volumes are library resident. The following changes are for EDGCNVT: <ul style="list-style-type: none"> • Physical volumes which are not library resident are marked in location SHELF. • Information about exported logical volumes and stacked volumes conversion is updated.
OA32574	The following changes are for CA-1 conversion: <ul style="list-style-type: none"> • Avoid ANDVRS usage with COUNT(0). • Information about exported logical volumes and stacked volumes conversion is added.
OA28637	Add REGION=32M to EDGHSKP step of EDGJVTC2.
OA29681	The following changes are for CONTROL-M/Tape conversion: <ul style="list-style-type: none"> • Add conversion of EDM tapes. • Support conversion for RETAIN parameters specified via CTPARM DD.
OA28407	Add NOCHALT to SORT options.
OA24025	Add conversion of the CA tape encryption key index field for both CA-1 and CA TLMS conversions.
z/OS V1R12	5.1 Cleanup VRSEL(OLD).
OA25233	The following changes are for CONTROL-M/Tape conversion: <ul style="list-style-type: none"> • Support CONTROL-M/Tape release 6.2. • Correct conversion of a RULE priority. • Add conversion of abend retention type records. • Set density to IDRC value as indication of compaction of data.
OA25672	TLMS conversion for stacked volumes. <ul style="list-style-type: none"> • Update EDGCDYNM to handle stacked volumes and exported logical volumes. • Update EDGCNVT to set the default owner for stacked volumes.
OA24896	Some minor change has been done in EDGCVTC4.
OA25112	Add new 64 bits fields: <ul style="list-style-type: none"> • Correct that some values (Block Count, Total Block Count, Data Set Size, Volume Usage) may be set to zero during conversion to DFSMSrmm.
OA24098	Add CDSID definition: <ul style="list-style-type: none"> • Add the RMMCDS statement to the list of EDGCNVT SYSIN statements. • Add RMMCDS EQUALS MYCDSID to SYSIN DD of EDGJCNVT sample JCL. • Add CDSID(MYCDSID) to SYSIN DD of EDGJVERR sample JCL.
OA24203	The following change is for ASG-ZARA/AutoMedia conversion: <ul style="list-style-type: none"> • Support ASG-ZARA release 1.7.

Table 1. Summary of changes (continued)

APAR / Development	Description
OA23102	<p>The following changes are for TMS conversion:</p> <ul style="list-style-type: none"> • Avoid creating duplicate CATALOG NAME VRS. • Add new media support to EDGC5BIN/EDGJ5BIN. • Correct message EDGC5BIN-00020. • Correct K-records initialization in EDGCSVDS.
OA20361	<p>The following changes are for ASG-ZARA/AutoMedia conversion:</p> <ul style="list-style-type: none"> • Correct conversion of a volume expiration date for multi data set volumes. • Recognize 1999/365 special expiration date as a permanent retention. • Change conversion of DMGE tapes from SCRATCH to MASTER pending replace. • Generate a next VRS name as XDnnnnny instead of XDYnnnnn (y is alphabetic character A-Z) to avoid duplicated VRS names for different locations. • Additionally, EDGCNVT is corrected so that IF VOLRANGE control statement does not override a volume location where the volume is offsite.
OA13370	Vendor specific media information support.
OA17574	Key encryption key labels and key encoding mechanisms (1 and 2) added to EDGCLREC - volume data.
OA17486	<ul style="list-style-type: none"> • Add TLMS retention types A,B,C processing. Handle like types 1,2,3 and issue message saying RETAINBY(SET) needs to be set. • Correct messages text alignment, conversion of data set create unit and record format. • Add conversion of last unit for first and second data set.
OA14918	Correct total block count setting (CONTROL-M/Tape).
OA13591	<ul style="list-style-type: none"> • CA-1 release 11.0 support. • Correct 'bad arithmetic conversion' in EDGRCTT2. • Ensure that volume expiration date is correct during TLMS conversion. • Support retention period 'RP PERMANENT' CA-1 parmlib option. • Add recognition of EXPDT=90000 special expiration date and create CATALOG VRS during CA-1 conversion. • Remove old conversion tools references.

Chapter 1. Before You Begin

Beginning with z/OS release V2R1, some DFSMSrmm samples and modules used by the conversion instructions in this guide have been removed from SYS1.SAMPLIB and SYS1.LINKLIB. Use the V1R13 sources of these modules instead.

Use the JCL provided on the z/OS downloads page <http://www-03.ibm.com/systems/z/os/zos/tools/downloads/index.html> to generate the removed modules from the sources. If you don't have the V1R13 sources available, contact your IBM service representative to get them.

You can open **EDGCMM01** on SYS1.SAMPLIB for a list of all parts, previously shipped in SYS1.SAMPLIB. Load the parts into a PDS source library of your choice on your z/OS system, where the RMM conversion will take place. Make sure to tailor the SET statement of various sample JCLs to use the correct source library. If you are currently licensed to z/OS and accept the terms according to the disclaimer below, you may use the sample code.

If you do not agree with these terms or you do not currently have a license to IBM z/OS, DO NOT USE THE SAMPLE CODE.

Disclaimer

SET FORTH BELOW ARE IMPORTANT TERMS REGARDING THE AGREEMENT FOR "AS IS" USE OF SAMPLE DFSMSRMM CONVERSION CODE ("SAMPLE CODE"). YOU MUST READ THESE TERMS BEFORE DOWNLOADING, INSTALLING OR USING THIS SAMPLE CODE. ANY DOWNLOAD, INSTALLATION OR USE INDICATES YOUR ACCEPTANCE OF THESE TERMS.

Sample Code is available only to customers who have a license to IBM z/OS ("Qualified Customers"). If you do not have a license to z/OS, you may NOT download the Sample Code.

If you are a Qualified Customer, you are hereby granted a nonexclusive, royalty-free license to install and use the Sample Code on computer systems which contain a licensed copy of z/OS.

This license also includes the right to make backup copies of the Sample Code. All other rights, title, and interest in and to the Sample Code are not transferred by this license and remain with the author and IBM.

This license ends when you no longer have a licensed copy of z/OS.

THE SAMPLE CODE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF PATENTS, COPYRIGHTS OR OTHER PROPRIETARY RIGHTS OF OTHERS.

NEITHER THE AUTHOR NOR IBM WARRANT THAT THE FUNCTIONS, ROUTINES, AND DATA CONTAINED IN, OR GENERATED AS A RESULT OF THE SAMPLE CODE, WILL MEET YOUR REQUIREMENTS OR BE ERROR-FREE.

The entire risk related to the quality and performance of the Sample Code is with you. In the event that there is any defect, you assume the entire cost of all necessary services, repair or correction.

IN NO EVENT WILL THE AUTHOR OR IBM BE LIABLE TO YOU OR TO ANY THIRD PARTY FOR ANY DIRECT OR INDIRECT DAMAGES (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES) ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM, EVEN IF THE AUTHOR OR IBM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Some jurisdictions do not allow the exclusion of implied warranties or the limitation or exclusion of liability for incidental or consequential damages, so some of the above may not apply to you.

This agreement does not entitle you to any consulting, program maintenance or other services.

For general conversion process questions please refer to:

- Converting to DFSMSrmm from CA-1 (SG24-6241)
- Converting to DFSMSrmm from TLMS (SG24-6242)
- Converting to DFSMSrmm from Control-T (SG24-6243)
- Converting to DFSMSrmm from ICF-usercatalog (SG24-6244)
- Converting to DFSMSrmm from ASG-ZARA or AutoMedia (SG24-6275)
- Converting to Removable Media Manager A Practical Guide (SG24-4998)

You can check for the DFSMSrmm conversion Redbooks on <http://www.redbooks.ibm.com> and search for 'RMM'.

For the most recently updated information about samples and programs please refer to this User's Guide.

Future updates will include changes which are additions to the information in the Redbooks - so you should use both the Redbooks and this document to ensure that you have the best information about the conversion tools.

This documentation reflects the current maintenance level of RMM, and you should always use the documentation level that reflects your current maintenance. You can check the level in SAMPLIB to ensure that you are using the level that matches the level of the conversion programs you are using.

For the latest product documentation refer to:

- "**z/OS DFSMSrmm Implementation and Customization Guide**" (SC26-7405) and
- "**z/OS DFSMSrmm Managing and Using Removable Media**" (SC26-7404)

For changes made to the product since that book level refer to edgdoc.pdf.

The provided sample programs and this User's Guide are very general. Your knowledge of your installation's policies and operating environment is the key to the success of the conversion process. In some instances, it may be necessary to modify the provided sample programs to achieve the required results.

The sample extract programs and JCLs, and the EDGCNVT utility can, and have, been used to convert data from a number of different systems. You will need to modify such things as:

- Data Set names
- Data Set sizes
- Execution parameters
- Control files
- | • Source library name, to where you have uploaded the sources from the web
- | • Target library name, to where you compile and link the load modules, which must reside in your
- | linklib concatenation

It is possible that you use an option of your existing management system that is not taken into account by the sample extract programs. In those cases you will need to update the sample programs to provide the function you require.

The following samples are provided:

Table 2. DFSMSrmm-supplied conversion samples

Name	Location	Function
HLASM modules		

Table 2. DFSMSrmm-supplied conversion samples (continued)

Name	Location	Function
EDGCNVT	LINKLIB	The EDGCNVT program
EDGCEBIN	LINKLIB	The EDGCEBIN program
EDGCVOVL	LINKLIB	The EDGCVOVL program
EDGCNXIT	SAMPLIB	Source for sample exit for EDGCNVT
EDGCDYNM	WEB	Source for TLMS extract program
EDGCSVDS	WEB	Source for TMS extract program
EDGC5BIN	WEB	Source for TMS 5.0 and above extract program
EDGC5LDR	WEB	Source for TMS 5.0 and above extract program
EDGCRFMT	WEB	Source for TMS 5.0 and above extract program
EDGCSRDS	WEB	Source for TMS extract program
EDGCVRSX	WEB	Sample EDGUX100 exit
EDGCVRSXG	WEB	Macro used for EDGCVRSX/F
EDGCVRSF	WEB	Part of sample EDGUX100 table
EDGCVRSE	WEB	Part of sample EDGUX100 table
EDGCVRSL	WEB	Sample to load EDGUX100 table
EDGC1PRM	WEB	Parameters Extract Program From TMS Parmlib
EDGC5UJL	WEB	Convert Data Sets Using Unique Julian Date
EDGCSPCC	WEB	CA1: Special Character Conversion
EDGC1PPL	WEB	Maps the TMS Parameter Area
REXX execs		
EDGRACS	WEB	Exec for TMS to create MC ACS statements
EDGRACSCR	WEB	Exec for TMS 5.0 and above pool conversion
EDGCVTC0	WEB	Create LISTCAT commands by using the LISTCAT UCAT command
EDGCVTC4	WEB	Create ADDRACK, ADDVOLUME and CHANGEVOLUME commands to build an RMM CDS from the ICF-CATALOG input
EDGCVTC5	WEB	Create additional ADDVOLUME and CHANGEVOLUME commands
EDGCVTC6	WEB	Create CHANGEVOLUME commands to build the volume chains
EDGRT2EX	WEB	Source for TAPE2000 extract exec
EDGRCTT1	WEB	REXX source to create the extended Control-M/Tape record
EDGRCTT2	WEB	REXX source for Control-M/Tape media data base (MDB) extract
EDGRCTT3	WEB	REXX source for Control-M/Tape Rules extract
EDGRCZA1	WEB	REXX source to modify the unloaded ZARA/AutoMedia records for future processing
EDGRCZA2	WEB	REXX source to create L-, D-, K-, and O-Records for the EDGCNVT conversion program
EDGRCZA3	WEB	REXX source to create K-Records out of a ZARA/Automedata expiration report
EDGCLMS	WEB	Sample Homegrown Conversion CLIST
JCLs		
EDGJCNVT	SAMPLIB	JCL to execute EDGCNVT and EDGCEBIN
EDGJVERR	SAMPLIB	JCL to create a CDS control record and execute EDGUTIL VERIFY processing

Table 2. DFSMSrmm-supplied conversion samples (continued)

Name	Location	Function
EDGJLOAD	SAMPLIB	JCL to execute EDGCVOVL, create and load CDS
EDGJYPRE	WEB	JCL to compile and link EDGCDYNM
EDGJDYNM	WEB	JCL to execute EDGCDYNM
EDGJ5PRE	WEB	JCL to compile and link EDGCRFMT, EDGCSPCC, EDGC5UJL, EDGC1PRM, EDGCSRDS, EDGCSVDS, EDGC5BIN, EDGC5LDR, EDGCVRSL, EDGCVRSX
EDGJSRDS	WEB	JCL to execute EDGCSRDS and EDGRACS
EDGJSVDS	WEB	JCL to execute EDGCSVDS and reformat the RDS and VPDD
EDGJ5BIN	WEB	JCL to execute EDGC5BIN
EDGJ5LDR	WEB	JCL to execute EDGC5LDR and reformat the RDS and VPDD
EDGJCSCR	WEB	JCL to execute exec EDGRCSCR
EDGJVTC1	WEB	JCL to convert from manual management part 1
EDGJVTC2	WEB	JCL to convert from manual management part 2
EDGJVTC3	WEB	JCL to convert from manual management part 3
EDGJT2EX	WEB	JCL to execute EDGRT2EX TAPE2000 conversion exec
EDGJCTT1	WEB	JCL to execute EDGRCTT1 CONTROL-M/Tape conversion exec
EDGJCTT2	WEB	JCL to execute EDGRCTT2 CONTROL-M/Tape conversion exec
EDGJCTT3	WEB	JCL to execute EDGRCTT3 CONTROL-M/Tape conversion exec
EDGJCZA1	WEB	JCL to execute EDGRCZA1 ASG-ZARA/AutoMedia conversion exec
EDGJCZA2	WEB	JCL to execute EDGRCZA2 ASG-ZARA/AutoMedia conversion exec
EDGJCZA3	WEB	JCL to execute EDGRCZA3 ASG-ZARA/AutoMedia conversion exec
EDGCVRSP	WEB	Procedure to run EDGCVRSL

The sample JCL is setup to work in a 'typical' installation. There is sample JCL, some of which is commented out so that you choose to use it or not as is your preference.

Note: Make sure to run EDGJ5PRE or EDGJYPRE prior to all other sample JCLs, in order to compile and link the modules required for the extraction process.

Chapter 2. Introduction

The conversion process is really a three phase methodology. Of course there is a planning effort that needs to happen prior to any conversion. You may find it very useful and informative to execute the three conversion phases, even during the planning phase, because the data extraction programs in particular can tell you a lot about what exists in your existing tape management system environment, what will and will not fit into the RMM environment, and expose problems that will need to be solved before moving to RMM in production mode.

The first phase is the Data Extraction Phase where the data that is needed by the RMM product is derived from the customer's current database. This User's Guide, although it will address all three phases of the process, is designed to assist you primarily with the first phase.

The second phase is the manipulation of the extracted data into a format that is, when loaded into a VSAM KSDS, useful to the RMM product. It is the EDGCNVT program together with the EDGCVOVL and EDGCEBIN programs that accomplish phase two, called the Convert Phase.

The third phase, referred to as the Postprocessing Phase, is the loading of the VSAM KSDS, the addition of the RMM CDS Control Record, and the utility run that cross checks the records in the RMM database, one against the other, to verify completeness and accuracy. The addition of the RMM Control Record and the Verification of the database are the function of the EDGUTIL program.

For details of planning and executing the conversion to DFSMSrmm refer to the list of Redbooks, referred above.

Chapter 3. Conversion from TMS/CA-1

This chapter contains information about converting from TMS/CA-1 to DFSMSrmm. For general conversion process questions refer to the Redbook "**Converting to DFSMSrmm from CA-1**" (SG24-6241). If you do not have it, we recommend you to order a copy. You can check for the Redbooks on: <http://www.redbooks.ibm.com> and search for 'RMM'.

The following steps summarize the recommended conversion process:

1. Review
 - "Converting to DFSMSrmm from CA-1"
 - the rest of this chapter
 - and Chapter 9, "Building the DFSMSrmm CDS," on page 185
2. Run TMSDATA to produce the data required for conversion. If you are running CA-1 5.1 or above you can replace TMSDATA with the use sample SORT JCL in the EDGJ5BIN job. You will have to remove the comments from the JCL in order to use this alternative. Run EDGCJCSR to convert TMS/CA-1 pool definitions and rules.
3. Run EDGC5BIN with the CHECK execution parameter to determine outcode locations in use.
4. Evaluate existing retention management and volume movement policies.
5. Formulate EDGCxxxx and EDGCNVT control statements.
6. Define and customize the RMM system.
7. Start RMM subsystem in MANUAL mode to get familiarization with dialog and commands.
8. Execute the programs required for extract and conversion:
 - a. EDGRCSR
 - b. EDGC5BIN
 - c. EDGC5LDR
 - d. EDGC5BIN (optional)
 - e. EDGCSRDS
 - f. EDGRACS
 - g. EDGCSVDS
 - h. EDGCNVT
 - i. EDGCEBIN
9. Load the RMM Control Data Set.
10. Use EDGUTIL to verify the RMM control data set and create the control record.
11. Tailor the VLPOOL definitions created by EDGRCSR and optionally merge the Storage Group and Management Class assignment statements from EDGRCSR with your ACS routines.
12. Check, update and merge the ACS statements generated by EDGRACS with existing ACS routines, according to your retention policies. Define the new management classes. Activate the ACS routines and run the RMM ADDVRS sub-commands created by EDGRACS. Another way to implement the retention policies is by using the UXTABLE. The recommendation is to use the ACS routines.
13. Start RMM subsystem in RECORD mode.
14. Manually correct any duplicate or GDG VRS definitions and save changes.
15. Execute RMM inventory management process (VRSEL, EXPROC and DSTORE) to assign storage locations and bin numbers.
16. Compare RMM scratch and storage location reports with corresponding TMS reports.
17. Correct any conversion problems and repeat steps 8 to 17.

18. Modify and install any required installation exits, such as EDGUX100.
19. Start RMM in WARNING mode.
20. When no unexplained differences, start RMM in PROTECT mode and remove TMS from the system.

Please refer to Table 2 on page 3 for a list of supplied conversion samples.

Preparing to run the Extract

The data extraction programs extract the proper information from the TMC volume records and DSNB records and convert the information about retention and vaulting contained in the RDS and VPDD.

Use the vendor-provided utility, TMSDATA, to unload volume records and DSNB records from the TMC database (see Table 7 on page 31 for detailed information about the processing requirements of the TMC volume records and the DSNB records).

If TMSDATA is not available, see sample EDGJ5BIN for the JCL to extract the records using DFSORT Release 11.1 or higher.

Before executing TMSDATA, you must ensure that there are no broken volume chains or data set chains. TMS/CA-1 provides the following utilities to accomplish this function:

- TMSPTRS (at the 9308 level or higher for 5.0)
- TMSFVSN

In addition, check that the expiration dates used under 5.0 or later levels of CA-1 are not set to STATS/xxx. CA-1 treats these dates as permanent retention, but you should correct them to be valid expiration dates before conversion. If you do not, the extract programs issue an information message and set a permanent retention date.

In addition, consider running other CA-1 utilities to ensure that the data being converted to RMM is current and will probably remain so long enough to complete the conversion process and validate the results.

For example, consider running the following:

TMSEXPDT

to set the correct expiration dates for volumes on the basis of your RDS definitions and CA-1 parameters.

TMSCTLG

to remove volumes from cycle control.

TMSCLEAN

to return volumes to scratch status.

TMSVMXX

to make retention decisions based on the VPDD and assign volumes to vaults and vault slots.

The following figure shows a sample DFSORT JCL:

```

//CVT50SRT JOB ,140.SCHLUMBERGER,MSGCLASS=H,REGION=6M,
//      MSGLEVEL=(1,1),NOTIFY=SCHLUM
//*****
//* SPLIT CA-1 DATABASE *
//*   PRODUCES SEQUENTIAL DATA SETS FROM TMC VOLUME RECORDS AND *
//*   DATA SET NAME BLOCKS (DSNB) RECORDS *
//* *
//* PREREQUISITE DFSORT REL. 11.1 OR REL. 11.0 INCLUDE THE *
//*   APAR PL82191 *
//* *
//* CA-1 REL. 5.0 or higher *
//*   TMC CONTROL RECORDS *
//*   EXCLUDING THE FIRST 3 OR 4 CA-1 CONTROL RECORDS *
//*   TMC VOLUME RECORD *
//*   SORTED BY FIRST VOLSER, VOL SEQ AND VOLSER *
//*   1. FIRST VOLSER OF DATASET      OFFSET 56  LENGTH 6 *
//*   2. VOLUME SEQUENCE NUMBER      OFFSET 54  LENGTH 2 *
//*   3. VOLUME SERIAL NUMBER        OFFSET  0  LENGTH 6 *
//*   DSNB DATA SET RECORD *
//*   1. FIRST VOLSER NUMBER OF THE 1ST. OFFSET 16  LENGTH 6 *
//*   2. FILE SEQUENCE NUMBER        OFFSET 28  LENGTH 2 *
//* *
//*****
//DELETE EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN DD *
DELETE DFRMM.CUSTOMER.CA1.VOLUME1 NONVSAM SCRATCH
DELETE DFRMM.CUSTOMER.CA1.DSNB1 NONVSAM SCRATCH
/*

```

Figure 1. DFSORT: Sample JCL (Part 1 of 4)

```

//SORT01 EXEC PGM=SORT,REGION=6M
//SYSPRINT DD  SYSOUT=*
//SORTIN DD  DISP=SHR,DSN=DFRMM.CUSTOMER.CA1
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTOUT DD  DSN=DFRMM.CUSTOMER.CA1.VOLUME1,DISP=(,CATLG,DELETE),
//          DCB=(RECFM=FB,LRECL=340,BLKSIZE=0),UNIT=SYSDA,
//          SPACE=(27880,(160),RLSE,,ROUND)
//SYSOUT DD  SYSOUT=*
//* -----
//*   TMC VOLUME RECORDS
//*   SORTED BY FIRST VOLSER, VOLUME SEQUENCE AND VOLSER
//*   EXCLUDE TMC CONTROL RECORDS, DSNB RECORDS AND
//*   VOLUME IN DELETE (INACTIVE) STATUS
//* -----
//SYSIN DD *
OPTION NOCHALT
SORT FIELDS=(57,6,CH,A,55,2,BI,A,1,6,CH,A)
OMIT COND=(1,7,CH,EQ,C'TMSCTL#',
           OR,1,1,CH,EQ,X'FF',
           OR,85,1,BI,EQ,B'.....1.')
/*

```

Figure 1. DFSORT: Sample JCL (Part 2 of 4)

```

//SORT02 EXEC PGM=SORT,REGION=6M
//SYSPRINT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=DFRMM.CUSTOMER.CA1
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTOUT DD DSN=&&TEMP,DISP=(,PASS,DELETE),
//          DCB=(RECFM=FB,LRECL=340,BLKSIZE=0),UNIT=SYSDA,
//          SPACE=(27880,(900),RLSE,,ROUND)
//SYSOUT DD SYSOUT=*
//* -----
//*      DSNB DATA SET RECORD
//*      SORTED BY COMPLETE RECORD
//*      INCLUDE ONLY DSNB RECORDS
//* -----
//SYSIN DD *
OPTION NOCHALT
SORT FIELDS=(1,340,CH,A)
INCLUDE COND=(1,1,CH,EQ,X'FF')
/*

```

Figure 1. DFSORT: Sample JCL (Part 3 of 4)

```

//SORT03 EXEC PGM=SORT,REGION=6M
//SYSPRINT DD SYSOUT=*
//SORTIN DD DISP=(OLD,DELETE),DSN=&&TEMP,
//          DCB=(RECFM=FB,LRECL=170,
//          BLKSIZE=0)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTOUT DD DSN=DFRMM.CUSTOMER.CA1.DSNB1,DISP=(,CATLG,DELETE),
//          DCB=(RECFM=FB,LRECL=170,BLKSIZE=27880),UNIT=SYSDA,
//          SPACE=(27880,(900),RLSE,,ROUND)
//SYSOUT DD SYSOUT=*
//* -----
//*      DSNB DATA SET RECORD
//*      SORTED BY FIRST VOLSER AND FILE SEQUENCE
//*      INCLUDE ONLY DSNB RECORDS WITH DSNB USED INDICATOR
//* -----
//SYSIN DD *
OPTION NOCHALT
SORT FIELDS=(17,6,CH,A,29,2,BI,A)
INCLUDE COND=(1,2,CH,EQ,X'FF80')
//

```

Figure 1. DFSORT: Sample JCL (Part 4 of 4)

Note: Each input record of 340 bytes will be split into two single 170 bytes records. This means to split the left and right DSNB record into a single 170 byte length file.

The data extraction programs also convert installation-defined tape management policies (retention and outcode location movement) stored in the RDS and VPDD.

In addition, the data extraction programs create empty bin records to represent unassigned bins (slots) in VMS-managed vaults (see Figure 15 on page 45).

The data extraction programs create a series of sequential output data sets that become the input to the EDGCNVT program. The output from EDGCNVT is used to build the DFSMSrmm CDS.

The data extraction jobs and programs provide the following functions:

- The EDGRCSR exec converts scratch pool information from the CA-1 parmlib for use with EDGC5LDR and with SMS ACS and RMM.

- The EDGC5LDR program converts the TMC volume records and the DSNB records to L-Records and D-Records and creates K-Records for the special expiration dates supplied in the VRS management value (VRSMGMT DD) data set. It can optionally create empty bin records. It also creates a file of keyword dates for input to EDGCSRDS.
- The EDGC5BIN program creates E-Records representing the empty bins in each of the vaults. When run with the CHECK execution parameter it can produce a list of OUTCODES you are using and classify them as STORES (VAULTS) and LOAN (all other outcodes). Use these EDGC5BIN created lists as input to EDGC5LDR and EDGCSVDS.
- The EDGCSRDS program converts the RDS retention policies to K-Records. EDGCSRDS also converts the keyword dates and RDS values into a table for use with EDGUX100.
- The EDGRACS exec extracts retention policy information from the table created by EDGCSRDS and produces two files. The file contains SMS ACS statements. The second file contains RMM ADDVRS subcommands.
- The EDGCSVDS program converts the VPDD movement policies to K-Records.

| **Note: Make sure to run EDGJ5PRE prior to all other sample JCLs, in order to compile and link the**
| **modules required for the extraction process.**

Conversion of TMS/CA-1 Pool Definitions and Rules

This is an optional part of the conversion. Run it if you plan to use RMM with SMS ACS processing to select scratch pools for non-system managed tape data sets. Regardless of any plan to use SMS ACS routines, you can use this step to help create the RMM VLPOOL definitions for parmlib.

Input

The input definitions are from CA-1 parmlib members:

TMOSCRxx

The scratch pool definitions.

```
SCRPOOL=name,RANGE=nnnnnn-mmmmm,RANGE ...
```

Conversion tools use the SCRPOOL name value as the RMM pool name and volume storage group name. This means that SCRPOOL name values are limited to 8 characters maximum.

If any SCRPOOL name values are more than 8 characters you should copy the CA-1 parameters and edit the SCRPOOL name values to be within the limits for RMM and the conversion tools.

Note: Ranges with lowvolser (nnnnnn) and/or highvolser (mmmmmm) less than 6 characters will be left justified and padded with blanks.

TMONSMxx

The scratch pool assignment rules.

```
MGMTCLAS=class,POOL=name
EDM=edm,POOL=name
DSN=dsname,POOL=name
DSN=dsname,JOB=job,UNIT=unit,EXPDT/RETPD=xxxx,POOL=name
```

TMOEDMxx

The external data manager definitions.

```
EDM=edm,DSN=dsname,PGM=pgm,DD=ddname,JOB=job
```

The pool definitions are converted to:

1. RMM VLPOOL definitions.
2. Pool table for use by EDGC5LDR to set a storage group for each volume.

The pool rules are converted to SMS ACS routine logic for use in your management class and storage group ACS routines. The pool rules DSN, JOB, PGM, DD, and UNIT can be pattern masks.

Pattern masks are converted as follows:

!, #, @, ?

(exclamation, hash, at, question) are converted to % (percent). When ? is found also a warning message is issued because there is no exact equivalent.

* (asterisk) is converted to *.

- (dash) is converted to **, *.* , **.*, or *.*.* depending on the position of the mask character relative to preceding/following qualifiers and periods.

/ (slash) is deleted from masks and a warning message issued. The following non-mask character is left unaltered.

Any warning messages should be resolved by altering the input data set masks to meet DFSMS filter standards.

When the EDM= keyword is used the TMOEDMxx option file must be used to determine the selection criteria for the EDM pool rule.

EDGRCSCR - Process CA-1 Pool Definitions

Run EDGRCSCR using the sample JCL and your input files. See sample EDGJCSCR for the supplied sample JCL.

Check for any messages issued during processing and validate that the output produced will meet your needs running under RMM. If you have complex scratch pools with overlapping ranges, or pool names longer than 8 characters you can edit the input files to make adjustments to help the conversion tools provide improved output.

Do not change the live CA-1 files, but create copies for use just as input to the conversion tools.

EDGRCSCR REXX procedure reads all three input files and produces possible output. See Table 4 on page 16 for details of the output files.

```

//RCSCR   JOB
//   SET RMMLIB=xxxxxxx
//*
//* CLEANUP FOR RERUN
//*
//DEL     EXEC PGM=IEFBR14
//VLPOOL  DD DISP=(MOD,DELETE),SPACE=(TRK,1),UNIT=SYSALLDA,
//         DSN=STSGWD.EDGRCSR.VLPOOL.DATA
//MCLASS  DD DISP=(MOD,DELETE),SPACE=(TRK,1),UNIT=SYSALLDA,
//         DSN=STSGWD.EDGRCSR.MCLASS.DATA
//SGROUP  DD DISP=(MOD,DELETE),SPACE=(TRK,1),UNIT=SYSALLDA,
//         DSN=STSGWD.EDGRCSR.SGROUP.DATA
//SGNAME  DD DISP=(MOD,DELETE),SPACE=(TRK,1),UNIT=SYSALLDA,
//         DSN=STSGWD.EDGRCSR.SGNAME.DATA
//*
//* RUN THE EDGRCSR PROCEDURE TO CONVERT THE CA-1 SCRATCH POOLS AND
//* SCRATCH POOL ASSIGNMENT RULES FOR USE WITH DFSMSRMM
//*
//*
//TMP     EXEC PGM=IKJEFT01,PARM='%EDGRCSR'
//SYSPROC DD DISP=SHR,DSN=&RMMLIB
//SYSTSPT DD SYSOUT=*
//SYSTSIN DD DUMMY
//*****
//*** PLEASE SPECIFY THE NAME OF THE DATASET CONTAINING THE ***
//*** CA-1 SCRATCH POOL DEFINITIONS, SCRATCH POOL ASSIGNMENT ***
//*** RULES, AND EXTERNAL DATA MANAGER SPECIFICATIONS.      ***
//***                                                         ***
//*** NOTE: WHEN EDM=KEYWORD IS USED IN THE TMONSM FILE THEN ***
//*** THE TMOEDM FILE CONTAINING THE EXTERNAL DATA MANAGER ***
//*** SPECIFICATIONS MUST BE USED.                            ***
//*****
//TMOSCR  DD DSN=CAI.PPOPTION(TMOSCRXX),DISP=SHR
//TMONSM  DD DSN=CAI.PPOPTION(TMONSMXX),DISP=SHR
//TMOEDM  DD DSN=CAI.PPOPTION(TMOEDMXX),DISP=SHR
//VLPOOL  DD DSN=STSGWD.EDGRCSR.VLPOOL.DATA,DISP=(,CATLG),
//         UNIT=SYSALLDA,SPACE=(TRK,(1,1)),LRECL=80,RECFM=FB
//MCLASS  DD DSN=STSGWD.EDGRCSR.MCLASS.DATA,DISP=(,CATLG),
//         UNIT=SYSALLDA,SPACE=(TRK,(1,1)),LRECL=80,RECFM=FB
//SGROUP  DD DSN=STSGWD.EDGRCSR.SGROUP.DATA,DISP=(,CATLG),
//         UNIT=SYSALLDA,SPACE=(TRK,(1,1)),LRECL=80,RECFM=FB
//SGNAME  DD DSN=STSGWD.EDGRCSR.SGNAME.DATA,DISP=(,CATLG),
//         UNIT=SYSALLDA,SPACE=(TRK,(1,1)),LRECL=80,RECFM=FB

```

Figure 2. EDGJCSR: Sample JCL

Input data sets

EDGRCSR has four input data sets:

Table 3. EDGRCSR Input Data Sets

DDNAME	Description
TMOSCR	CA-1 scratch pool definitions.
STORETMONSM	CA-1 scratch pool assignment rules.
TMOEDM	CA-1 EDM definitions.
SYSPROC	Data set containing the EDGRCSR exec.

Output Data Sets

EDGRCSR has five output data sets:

Table 4. EDGRCSR Output Data Sets

DDNAME	Description
SYSTSPRT	Information and error messages from execution.
SGNAME	<p>A file for input to EDGC5LDR.</p> <p>This file lists tape ranges by pool name. Each record lists a single range and a pool name for EDGC5LDR to use to set storage group names for volumes.</p> <pre>sssssseeeeeepoolnamescrpoolname</pre> <p>where</p> <p>sssss is the starting volser</p> <p>eeeeee ending volser</p> <p>poolname pool name to be used for RMM</p> <p>scrpoolname CA-1 pool name</p>

Table 4. EDGRCSER Output Data Sets (continued)

DDNAME	Description
VLPOOL	<p>A file of RMM VLPOOL definitions.</p> <p>The VLPOOL definitions may require to be edited before you use them with DFSMSrmm.</p> <p>During processing the VLPOOL definitions are built as follows:</p> <p>PREFIX</p> <p>most specific prefix value for range.</p> <p>If consecutive ranges have the same scrpool name, they are consolidated to a single VLPOOL.</p> <p>A warning message is issued if there are multiple ranges assigned to the same prefix range. For example, 000105-003007 and 003008-039999 cause an overlap.</p> <p>DFSMSrmm pools must be on 10, 100, 1000 boundaries and have no overlap. Conversion processing attempts to resolve overlaps, but warning messages are issued when this is not possible. Overlaps can be solved either by moving volumes to different scratch pool ranges to remove the overlap or by using rack numbers that are different than volser.</p> <p>EDGCNVT provides a way to set different rack number prefixes by volume range. If there are any overlaps you must edit the VLPOOL definitions and set the correct SYSIN values for the EDGCNVT</p> <pre>IF VOLRANGE SET RACK PREFIX</pre> <p>DESCRIPTION</p> <p>Is set to the first and last volser range:</p> <pre>DESCRIPTION('000105-003007 ... 004008-039999')</pre> <p>EXPDTCHECK</p> <p>N</p> <p>Consider changing this value if you want to honour expiration date protected tape volumes defined to RMM.</p> <p>MEDIANAME</p> <p>CARTS</p> <p>You must change this value to match the media names generated during conversion using either EDGC5LDR or EDGCNVT SYSIN control statements.</p> <p>NAME The first 8 characters of the SCRPOOL value.</p> <p>This is the RMM scratch pool name and matches the CA-1 scratch pool name if the name is 8 characters or less. You may want to change the names if they have been truncated to 8 characters.</p> <p>When EDGC5LDR runs it uses the values from the SGNAME file to assign these values as the storage group name for each volume in the volume range. You can then use SMS ACS processing to take scratch pooling decisions for non-system managed tape volumes.</p> <p>RACF N</p> <p>Change this value based on your decision to use RACF TAPEVOL profiles for volumes in this pool.</p> <p>TYPE S</p> <p>Change this value to R if the range is not to be used as an RMM scratch pool.</p>

Table 4. EDGRCSRC Output Data Sets (continued)

DDNAME	Description
MCLASS	A file containing ACS routine logic and filter lists for management class processing. If you plan to use SMS ACS processing to assign policies to new data sets on non-system managed tape volumes you must merge these statements into your Management Class ACS routine.
SGROUP	A file containing ACS routine logic and filter lists for storage group processing. If you plan to use SMS ACS processing to assign specific scratch pools to new data sets on non-system managed tape volumes you must merge these statements into your Storage Group ACS routine, and merge the MCLASS statements into the Management Class ACS routine.

Return codes

Table 5. EDGRCSRC Return Codes

Return code	Description
0	Normal processing.
4	Warning message has been issued. Probably wrong input data. See joblog.
8	Error message has been issued. Wrong input data. See joblog.
12	Input/output error when reading/writing a file.

Messages

EDGRCSRC-00000 I/O ERROR WHEN READING THE TMOSCR FILE, RC=*rc*

Problem determination: where

rc EXECIO return code. See z/OS "TSO/E REXX Users Guide" (SA22-7791)

Severity: Error

Explanation: An error occurred during reading the TMOSCR file.

System action: The current input request fails.

User response: Check, correct, and rerun the job.

EDGRCSRC-00005 WARNING: SCRPOOL NAME *name* LONGER THAN 8 CHARACTERS WILL BE TRUNCATED TO 8 CHARACTERS. EDIT TMOSCR FILE.

Problem determination: where

name The name of CA-1 scratch tape pool

Severity: Warning

Explanation: SCRPOOL name values are limited to 8 characters maximum.

System action: EDGRCSRC truncates the pool name to 8 characters.

User response: Edit the SCRPOOL name values to be within the limits for DFSMSrmm and the conversion tools.

EDGRCSRC-00010 I/O ERROR WHEN WRITING THE TEMPORARY FILE, RC=*rc*

Problem determination: where

rc EXECIO return code. See z/OS "TSO/E REXX Users Guide" (SA22-7791)

Severity: Error

Explanation: An error occurred during writing the temporary file.

System action: The current output request fails.

User response: Check, correct, and rerun the job.

EDGRCSR-00015 I/O ERROR WHEN CLOSING THE TEMPORARY FILE, RC=*rc*

Problem determination: where

rc EXECIO return code. See z/OS "TSO/E REXX Users Guide" (SA22-7791)

Severity: Error

Explanation: Input/output error during closing the temporary file.

System action: The current I/O request fails.

User response: Report the error to the system programmer.

EDGRCSR-00020 I/O ERROR WHEN WRITING THE SORT CONTROL FILE, RC=*rc*

Problem determination: where

rc EXECIO return code. See z/OS "TSO/E REXX Users Guide" (SA22-7791)

Severity: Error

Explanation: An error occurred during writing the SORT control file.

System action: The current output request fails.

User response: Check, correct, and rerun the job.

EDGRCSR-00025 I/O ERROR WHEN READING INTERNAL SORTOUT FILE, RC=*rc*

Problem determination: where

rc EXECIO return code. See z/OS "TSO/E REXX Users Guide" (SA22-7791)

Severity: Error

Explanation: An error occurred during reading the internal SORTOUT file.

System action: The current input request fails.

User response: Check, correct, and rerun the job.

EDGRCSR-00030 I/O ERROR WHEN WRITING THE SGNAME FILE, RC=*rc*

Problem determination: where

rc EXECIO return code. See z/OS "TSO/E REXX Users Guide" (SA22-7791)

Severity: Error

Explanation: An error occurred during writing the SGNAME file.

System action: The current output request fails.

User response: Check, correct, and rerun the job.

EDGRCSR-00035 ERROR: VOLUME RANGES *range1* AND *range2* (SCRPOOL = *name*) ARE INTERSECTED. SEE TMOSCR INPUT FILE.

Problem determination: where

range1 A tape range from the specified CA-1 scratch tape pool

range2 A tape range from the specified CA-1 scratch tape pool

name The name of CA-1 scratch tape pool

Severity: Error

Explanation: Right margin of the first range is less or equal to left margin of the second range.

System action: EDGRCSER sets RC=8 and continues processing.

User response: Check the RANGE statements in the TMOSER file, correct it and rerun the job.

EDGRCSER-00040 I/O ERROR WHEN WRITING THE VLPOOL FILE, RC=*rc*

Problem determination: where

rc EXECIO return code. See z/OS "TSO/E REXX Users Guide" (SA22-7791)

Severity: Error

Explanation: An error occurred during writing the VLPOOL file.

System action: The current output request fails.

User response: Check, correct, and rerun the job.

EDGRCSER-00045 I/O ERROR WHEN CLOSING THE VLPOOL FILE, RC=*rc*

Problem determination: where

rc EXECIO return code. See z/OS "TSO/E REXX Users Guide" (SA22-7791)

Severity: Error

Explanation: Input/output error during closing the VLPOOL file.

System action: The current I/O request fails.

User response: Report the error to the system programmer.

EDGRCSER-00050 I/O ERROR WHEN READING TMONSM FILE, RC=*rc*

Problem determination: where

rc EXECIO return code. See z/OS "TSO/E REXX Users Guide" (SA22-7791)

Severity: Error

Explanation: An error occurred during reading the TMONSM file.

System action: The current input request fails.

User response: Check, correct, and rerun the job.

EDGRCSER-00055 I/O ERROR WHEN READING TMOEDM FILE, RC=*rc*

Problem determination: where

rc EXECIO return code. See z/OS "TSO/E REXX Users Guide" (SA22-7791)

Severity: Error

Explanation: An error occurred during reading the TMOEDM file.

System action: The current input request fails.

User response: Check, correct, and rerun the job.

EDGRCSER-00060 UNSUPPORTED KEYWORD IN TMONSM FILE: *keyword*

Problem determination: where

keyword Invalid keyword

Severity: Warning

Explanation: Invalid keyword has been encountered in the TMONSM file. The following keywords are supported: MGMTCLAS, EDM, POOL, DSN, JOB, UNIT, EXPDT, and RETPD.

System action: EDGRCSCR continues processing.

User response: Check the TMONSM file, correct it and rerun the job.

EDGRCSCR-00065 UNSUPPORTED KEYWORD IN TMOEDM FILE: *keyword*

Problem determination: where

keyword Invalid keyword

Severity: Warning

Explanation: Invalid keyword has been encountered in the TMOEDM file. The following keywords are supported: EDM, DSN, JOB, PGM, and DD.

System action: EDGRCSCR continues processing.

User response: Check the TMOEDM file, correct it and rerun the job.

EDGRCSCR-00070 UNSUPPORTED KEYWORD IN TMONSM FILE OR MANDATORY KEYWORD (MGMTCLAS, EDM, OR DSN) MISSING. SEE RECORD *record*

Problem determination: where

record Incorrect record of the TMONSM file

Severity: Warning

Explanation: Invalid keyword has been encountered in the TMONSM file or a mandatory keyword has been omitted in the TMONSM file. The following keywords are supported: MGMTCLAS, EDM, POOL, DSN, JOB, UNIT, EXPDT, and RETPD.

System action: EDGRCSCR continues processing.

User response: Check the TMONSM file, correct it and rerun the job.

EDGRCSCR-00075 WARNING: '?' HAS NO EQUIVALENT GENERIC, SO '%' IS USED. SEE *keyword=value* **IN TMONSM OR TMOEDM FILE**

Problem determination: where

keyword Keyword, can be DSN, JOB, UNIT, DD or PGM

value The value containing the pattern mask "?"

Severity: Warning

Explanation: The pattern mask "?" has no equivalent generic.

System action: "?" is replaced by "%".

User response: Check the TMONSM or TMOEDM file, correct it, if it is necessary, and rerun the job.

EDGRCSCR-00080 WARNING: '/' HAS NO EQUIVALENT GENERIC, HAS BEEN DELETED. SEE *DSN=value* **IN TMONSM OR TMOEDM FILE**

Problem determination: where

value The value containing the pattern mask "/"

Severity: Warning

Explanation: The pattern mask "/" has no equivalent generic.

System action: "/" is removed.

User response: Check the TMONSM or TMOEDM file, correct it, if it is necessary, and rerun the job.

EDGRCSR-00085 WARNING: '/' HAS NO EQUIVALENT GENERIC, HAS BEEN DELETED. SEE JOB=*value* IN TMONSM OR TMOEDM FILE

Problem determination: where

value The value containing the pattern mask "/"

Severity: Warning

Explanation: The pattern mask "/" has no equivalent generic.

System action: "/" is removed.

User response: Check the TMONSM or TMOEDM file, correct it, if it is necessary, and rerun the job.

EDGRCSR-00090 I/O ERROR WHEN WRITING THE MCLASS FILE, RC=*rc*

Problem determination: where

rc EXECIO return code. See z/OS "TSO/E REXX Users Guide" (SA22-7791)

Severity: Error

Explanation: An error occurred during writing the MCLASS file.

System action: The current output request fails.

User response: Check, correct, and rerun the job.

EDGRCSR-00095 WARNING: '/' HAS NO EQUIVALENT GENERIC, HAS BEEN DELETED. SEE UNIT=*value* IN TMONSM FILE

Problem determination: where

value The value containing the pattern mask "/"

Severity: Warning

Explanation: The pattern mask "/" has no equivalent generic.

System action: "/" is removed.

User response: Check the TMONSM file, correct it, if it is necessary, and rerun the job.

EDGRCSR-00100 WARNING: '/' HAS NO EQUIVALENT GENERIC, HAS BEEN DELETED. SEE DD=*value* IN TMOEDM FILE

Problem determination: where

value The value containing the pattern mask "/"

Severity: Warning

Explanation: The pattern mask "/" has no equivalent generic.

System action: "/" is removed.

User response: Check the TMOEDM file, correct it, if it is necessary, and rerun the job.

EDGRCSR-00105 WARNING: '/' HAS NO EQUIVALENT GENERIC, HAS BEEN DELETED. SEE PGM=*value* IN TMOEDM FILE

Problem determination: where

value The value containing the pattern mask "/"

Severity: Warning

Explanation: The pattern mask "/" has no equivalent generic.

System action: "/" is removed.

User response: Check the TMOEDM file, correct it, if it is necessary, and rerun the job.

EDGRCSCR-00110 WARNING: POOL NAME *name* LONGER THAN 8 CHARACTERS. WILL BE TRUNCATED TO 8 CHARACTERS. EDIT TMONSM FILE

Problem determination: where

name Pool name from a scratch pool assignment rule

Severity: Warning

Explanation: Pool name values are limited to 8 characters maximum.

System action: EDGRCSCR truncates the pool name to 8 characters.

User response: Edit the pool name values to be within the limits for DFSMSrmm and the conversion tools.

EDGRCSCR-00115 I/O ERROR WHEN CLOSING THE MCLASS FILE, RC=*rc*

Problem determination: where

rc EXECIO return code. See z/OS "TSO/E REXX Users Guide" (SA22-7791)

Severity: Error

Explanation: Input/output error during closing the MCLASS file.

System action: The current I/O request fails.

User response: Report the error to the system programmer.

EDGRCSCR-00120 I/O ERROR WHEN WRITING THE SGROUP FILE, RC=*rc*

Problem determination: where

rc EXECIO return code. See z/OS "TSO/E REXX Users Guide" (SA22-7791)

Severity: Error

Explanation: An error occurred during writing the SGROUP file.

System action: The current output request fails.

User response: Check, correct, and rerun the job.

EDGRCSCR-00125 I/O ERROR WHEN CLOSING THE SGROUP FILE, RC=*rc*

Problem determination: where

rc EXECIO return code. See z/OS "TSO/E REXX Users Guide" (SA22-7791)

Severity: Error

Explanation: Input/output error during closing the SGROUP file.

System action: The current I/O request fails.

User response: Report the error to the system programmer.

EDGRCSCR-00130 ERROR: POSSIBLE OVERLAP! SEE TMOSCR INPUT FILE. VOLUME RANGES *range1* AND *range2*

Problem determination: where

range1 A tape range from the specified CA-1 scratch tape pool

range2 A tape range from the specified CA-1 scratch tape pool

Severity: Error

Explanation: Right margin of the first range is less or equal to left margin of the second range.

System action: EDGRCSR sets RC=8 and continues processing.

User response: Check the RANGE statements in the TMOSCR file, correct it and rerun the job.

EDGRCSR-00135 ERROR IN THE INPUT/OUTPUT FILE

Severity: Error

Explanation: A physical error performing I/O to a file specified in the previous error message has been encountered.

System action: EDGRCSR fails with RC=12.

User response: Report the error to the system programmer.

EDGRCSR-00140 UNEXPECTED CODE

Severity: Error

Explanation: Unexpected code has been encountered.

System action: EDGRCSR fails.

User response: Report the error to the system programmer.

Extracting Data from the TMC Records using EDGC5LDR

EDGC5LDR converts the TMC volume records to L-Records representing volumes and their first data sets and the DSNB records to D-Records representing the second and subsequent data sets. For each private volume, it validates the data set name on the volume.

If it encounters an invalid data set name on a private volume (that is, a data set name beginning with x'00'), it writes an error message to the ERROR DD data set unless the volume is owned by an EDM identified through the EDMNAME DD data set.

You must not proceed with the conversion process until you resolve the error conditions reported in the ERROR DD data set.

Optionally, EDGC5LDR creates K-Records that ultimately become RMM VRSs. These K-Records are built from data supplied in the VRSMGMT DD data set and their names are placed in the VRS management value fields of the appropriate D-Records.

You can also code VRSMV=YES in the VRSMGMT file instead of coding any keyword dates, and conversion processing will treat all 9998ddd and 9999ddd expiration dates as keyword dates. See Table 9 on page 32 for detailed information.

Calls EDGC1PRM to obtain R9=nnn, RC=nnn and ABE=nnn from the CA-1 parmlib (CAI.PPOPTION).

K-records generated for keyword dates for catalog or cycles retention are built including a NEXTVRS(RC) or NEXTVRS(R9) and K-records generated for RC and R9 that specify EXTRADAYS COUNT(ABE).

- RC is extra days after cycles retention
- R9 is extra days after catalog retention
- ABE is the number of days to retain data sets closed during an abend

EDGC5LDR attempts to convert the media information into values that work with DFSMSrmm. For each volume a media name is assigned, and a media type and recording format determined and set.

The TRTCH value in the extracted data is used to determine the exact values to use.

EDGC5LDR can differentiate between IBM media such as 3590, 3592, and TS1120, and Sun StorageTek media such as Redwood, and 9840. For non-IBM media a media information name such as 'STK' is set.

For other media types, not currently supported, or to change what EDGC5LDR determines you can later use the EDGCNVT SYSIN file with the

```
IF VMEDIA
```

statement to set values based on the volume serial number. Support for non-IBM media will require that the MEDINF command is used in the DFSMSrmm parmlib.

EDGC5LDR can optionally assign a storage group name to each volume based on the CA-1 scratch pool name. Use this option, by including SGNAME DD using input from EDGRCSR, if you plan to use the SMS ACS routines to perform scratch pooling for non-system managed volumes.

EDGC5LDR can also verify the chaining integrity of data sets spanning multiple volumes, multiple data sets on a volume, and multiple data sets spanning multiple volumes. When EDGC5LDR detects a broken data set chain in the input data, it attempts to fix the error in working storage, and the attempt is reported in the ERROR DD data set.

However, when EDGC5LDR detects a broken volume chain, it does not attempt to fix the error. TMSPTRS and TMSFVSN must be used.

TMSDATA and EDGC5LDR must then be rerun until all chaining errors are removed.

Sample JCL

The figure below shows sample execution JCL for TMS/CA-1 Release 5.0 and above. See sample EDGJ5LDR for the supplied sample JCL.

Note: To help identify some of the specifics about your current TMS/CA-1 environment, execute EDGC5BIN with PARM='CHECK'. This creates files of the storage and loan locations which can be input to other conversion programs.

EDGC5BIN externalizes the following for you:

- Outcode location names currently in use to help you define which outcode locations become RMM storage locations and which become RMM loan locations.
- Volumes occupying outcode slot number zero, which is not compatible with RMM.
- Outcode slots with more than one volume assigned, which is also not compatible with RMM.

```
//* Sort by first VOLSER, volume sequence, and VOLSER
/*
//SORTTMC EXEC PGM=SORT,REGION=4M
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=STSGWD.TMS50.VOLRECS.DATA
//SORTOUT DD DISP=(,PASS),AVGREC=K,SPACE=(340,(50,10),RLSE),
//          DSN=&&SORTTMC,UNIT=SYSDA
//SYSIN DD *
OPTION NOCHALT
SORT FIELDS=(57,6,CH,A,55,2,BI,A,1,6,CH,A),DYNALLOD
/*
//* Sort by VOLSER which file 2 starts and file sequence number
/*
```

Figure 3. EDGC5LDR: Sample JCL, CA-1 Release 5.0 and above (Part 1 of 5)

```

//SORTDSNB EXEC PGM=SORT,REGION=4M,COND=(0,NE)
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=STSGWD.TMS50.DSNBRECS.DATA
//SORTOUT DD DISP=(,PASS),AVGREC=K,SPACE=(170,(50,10),RLSE),
//          DSN=&&SORTDSNB,UNIT=SYSDA
//SYSIN DD *
OPTION NOCHALT
SORT FIELDS=(17,6,CH,A,29,2,BI,A),DYNALLO
//*

```

Figure 3. EDGC5LDR: Sample JCL, CA-1 Release 5.0 and above (Part 2 of 5)

```

//DEL EXEC PGM=IEFBR14
//D1 DD DISP=(MOD,DELETE),UNIT=SYSALLDA,SPACE=(TRK,1),
//   DSN=STSGWD.EDGCXLDL.DEXTOUT.DATA
//D2 DD DISP=(MOD,DELETE),UNIT=SYSALLDA,SPACE=(TRK,1),
//   DSN=STSGWD.EDGCXLDL.VRSVALUE.DATA
//D3 DD DISP=(MOD,DELETE),UNIT=SYSALLDA,SPACE=(TRK,1),
//   DSN=STSGWD.EDGCXLDL.DEXTOUTK.DATA
//*

```

Figure 3. EDGC5LDR: Sample JCL, CA-1 Release 5.0 and above (Part 3 of 5)

```

//EDGC5LDR EXEC PGM=EDGC5LDR,REGION=4M
//*STEPLIB DD DISP=SHR,DSN=RMM.LOAD
//SYSUDUMP DD SYSOUT=*
//REPORT DD SYSOUT=*
//TMSPARM DD DISP=SHR,DSN=STSGWD.TMS50.PARMLIB.DATA
//SGNAME DD DISP=SHR,DSN=STSGWD.EDGRCSR.SGNAME.DATA
//VRSVALUE DD DISP=(,CATLG),LRECL=80,RECFM=FB,DSORG=PS,
//          AVGREC=U,SPACE=(TRK,(1,1),RLSE),UNIT=SYSALLDA,
//          DSN=STSGWD.EDGCXLDL.VRSVALUE.DATA
//ERROR DD SYSOUT=*
//STORE DD DISP=SHR,DSN=STSGWD.EDGCXBIN.STOREOUT.DATA
//*STORE DD *
//* cols 1-4 Current storage location name
//*      5-12 Owner Id (optional)
//*      13-20 New storage location name
//LOAN DD DISP=SHR,DSN=STSGWD.EDGCXBIN.LOANOUT.DATA
//*LOAN DD *
//* cols 1-4 Loan location name
//*      5-12 Owner Id (optional)
//VRSMGMT DD *
VRSMV=YES
//* cols 1-5 Special meaning expiration date
//*      6-13 VRS management value name
//EDMNAME DD *
DFHSM
//* cols 1-8 EDM job name or EDMID
//*      9-16 EDM RACF ID (optional)
//*      17-21 EDM option ( NOEDM or EDM )
//TMC DD DISP=(OLD,DELETE),DSN=&&SORTTMC
//DSNB DD DISP=(OLD,DELETE),DSN=&&SORTDSNB
//DEXTOUT DD DISP=(,CATLG),UNIT=SYSALLDA,LRECL=700,
//          RECFM=VB,SPACE=(500,(200,100),RLSE),AVGREC=K,
//          DSN=STSGWD.EDGCXLDL.DEXTOUT.DATA
//DEXTOUTK DD DISP=(,CATLG),UNIT=SYSALLDA,LRECL=700,
//          RECFM=VB,SPACE=(500,(200,100),RLSE),AVGREC=K,
//          DSN=STSGWD.EDGCXLDL.DEXTOUTK.DATA

```

Figure 3. EDGC5LDR: Sample JCL, CA-1 Release 5.0 and above (Part 4 of 5)

```

//*
//* Step sorts the keyword dates in VRSVALUE file
//*
//SORTKEYS EXEC PGM=SORT,REGION=6M
//SYSPRINT DD SYSOUT=*
//SORTIN DD DISP=OLD,DSN=STSGWD.EDGCXLDL.VRSVALUE.DATA
//SORTOUT DD DISP=OLD,DSN=STSGWD.EDGCXLDL.VRSVALUE.DATA
//SYSOUT DD SYSOUT=*
//SYSIN DD *
OPTION NOCHALT
SORT FIELDS=(1,5,CH,A),DYNALLOC
END

```

Figure 3. EDGC5LDR: Sample JCL, CA-1 Release 5.0 and above (Part 5 of 5)

Execution Parameters

```

//EDGC5LDR EXEC PGM=EDGC5LDR,PARM='x/y'

```

Figure 4. EDGC5LDR Execution Parameter Syntax

In the syntax, x is one of the following execution parameters:

REASSIGN

requests that:

- Volumes assigned to specific outcode location names have their names translated, name for name, to RMM storage location names.
- Those volumes' assigned slot numbers be ignored and new bin numbers be generated.

CONSOLIDATE

requests that:

- Volumes residing in multiple outcode locations be combined into fewer RMM storage locations.
- Those volumes' slot numbers be ignored and new bin numbers be generated.

and *y* is the number of empty bins requested. You must create empty bin records to provide space for RMM to move additional volumes to the storage locations after conversion.

With this method you do not have to add the empty bin numbers using RMM ADDBIN commands.

Note: When CONSOLIDATE or REASSIGN is requested, EDGC5LDR builds the necessary empty bin records and thus eliminates the need to execute EDGC5BIN.

When creating the empty bin records EDGC5LDR sets the media name to CART3480. If you use subvaulting and plan to do the same under DFSMSrmm, you may have to modify EDGC5LDR to set the media names you want to use for each location. If CART3480 is not the value you want to use, update EDGC5LDR to set a suitable value for each location.

The execution options are provided to help in the following situations:

- Volumes assigned to slot 0.
If this is the only situation to be resolved, it is probably easier to force CA-1 to move the volumes to a nonzero slot in the outcode location.
- Multiple volumes per slot number.
If only a few slots are occupied by more than one volume, consider moving the necessary volumes to alleviate the situation.
- Total number of outcode locations to be converted to RMM storage locations is greater than the number you want to use under DFSMSrmm.

Note: Do not use any of the preceding options if any outcode location depends on an accurate outcode slot number.

The following figure shows a sample consolidation of two outcode locations to LOCAL that also creates 100 empty bin records for each RMM storage location.

```
//EDGC5LDR EXEC PGM=EDGC5LDR,PARM='CONSOLIDATE/100'  
//STORE DD *  
VLT1 LOCAL  
VLT2 LOCAL  
VLT3 REMOTE  
VLT4 DISTANT
```

Figure 5. EDGC5LDR Execution Parameters: Sample 1

Next figure shows a sample that reassigns bin numbers, translates the outcode location names to RMM storage location names, and creates 200 empty bin records in each RMM storage location.

```
//EDGC5LDR EXEC PGM=EDGC5LDR,PARM='REASSIGN/200'
//STORE DD *
VLT1 LOCAL
VLT2 REMOTE
VLT3 DISTANT
```

Figure 6. EDGC5LDR Execution Parameters: Sample 2

The next figure shows a sample that reassigns bin numbers (without translation of outcode location names to storage location names). The translation of the outcode location names will have to be done later through EDGCNVT as shown in Figure 8.

Note: EDGCNVT can neither consolidate nor ensure that not more than one volume will occupy any given bin.

The sample also shows how to create 200 empty bin records in each storage location.

```
//EDGC5LDR EXEC PGM=EDGC5LDR,PARM='REASSIGN/00000200'
//STORE DD *
VLT1
VLT2
VLT3
```

Figure 7. EDGC5LDR Execution Parameters: Sample 3

```
//EDGCNVT EXEC PGM=EDGCNVT
//SYSIN DD *
IF STORLOC EQUALS VLT1 THEN RMMSTORE EQUALS LOCAL
IF STORLOC EQUALS VLT2 THEN RMMSTORE EQUALS REMOTE
IF STORLOC EQUALS VLT3 THEN RMMSTORE EQUALS DISTANT
```

Figure 8. EDGCNVT Storage Location Name Reassignment

Note: The STORLOC keyword must start in column 4, and the outcode location name must start in column 19. The RMMSTORE keyword must start in column 33, and the RMM storage location name must start in column 49.

Required Data

A number of fields have been set to default values within EDGC5LDR (see Table 6). You have the option of modifying the defaults in the EDGC5LDR assembler source or waiting until the execution of the EDGCNVT program.

It may be easier to wait until execution of the EDGCNVT program, when the changes can be driven through the SYSIN DD data set.

Table 6. EDGC5LDR Modifiable Fields

Field	Current value	Description
HOMLOC	SHELF	Home location name
DFLTOWN	LIBRARY	Owner name
TAPE1600	TAPE1600	Media name
TAPE6250	TAPE6250	Media name
CART3480	CART3480	Media name
CART3490	CART3490	Media name

Table 6. EDGC5LDR Modifiable Fields (continued)

Field	Current value	Description
CART349E	CART349E	Media name
CART3590	CART3590	Media name
REDWOOD	REDWOOD	Media name
CART9840	CART9840	Media name
STKMEDIA	STK	Media information name

Note: The media names specified must match the RMM VLPOOL and LOCDEF definitions in the EDGRMMxx member of PARMLIB. If they do not match, errors will occur when RMM is active.

The next figure shows an example of how to change the default media values to your installation definitions through the SYSIN DD data set for the EDGCNVT program.

The sample shows how to change the pool names:

- From TAPE1600 to OLDREEL
- From TAPE6250 to REELS
- From CART3480 to SQUARE
- From CART3490 to BGSQUARE

```
IF VMEDIA EQUALS TAPE1600 THEN MEDIANAME EQUALS OLDREEL
IF VMEDIA EQUALS TAPE6250 THEN MEDIANAME EQUALS REELS
IF VMEDIA EQUALS CART3480 THEN MEDIANAME EQUALS SQUARE
IF VMEDIA EQUALS CART3490 THEN MEDIANAME EQUALS BGSQUARE
```

Figure 9. EDGCNVT SYSIN Control Statement Format for VMEDIA

Note: Any volumes defined in the TMC but never used for data, are defined to TMS as tape reels. As a result, the conversion programs can convert a volume with the wrong media name value. You can correct this before conversion by updating the TMCDEN or TMCTRTCH fields, or during conversion by using the EDGCNVT SYSIN statement

```
IF VOLRANGE ... THEN MEDIANAME EQUALS ...
```

EDGC5LDR now uses the TMC last change user ID as the volume owner, and only uses DFLTOWN when there us no last change user ID.

The following figure shows an example of how to change the owner name from LIBRARY to OPERATOR and add more information relative to the default owner through the SYSIN DD data set for the EDGCNVT program.

```
IF OWNER EQUALS 'LIBRARY ' THEN NEWOWNR EQUALS 'OPERATOR'
IF OWNER EQUALS 'LIBRARY ' THEN NEWOWNS EQUALS 'DATA CENTER'
IF OWNER EQUALS 'LIBRARY ' THEN NEWOWNF EQUALS 'COMPUTER ROOM'
IF OWNER EQUALS 'LIBRARY ' THEN NEWOWND EQUALS 'DEPARTMENT 999'
IF OWNER EQUALS 'LIBRARY ' THEN NEWOWNT EQUALS '999-9999'
IF OWNER EQUALS 'LIBRARY ' THEN NEWOWNU EQUALS 'OPERID '
IF OWNER EQUALS 'LIBRARY ' THEN NEWOWNN EQUALS 'SYSTEMA '
```

Figure 10. EDGCNVT SYSIN Control Statement Format for OWNER

Input Data Sets

EDGC5LDR has eight input data sets:

Table 7. EDGC5LDR Input Data Sets

DDNAME	Description	Usage
TMC	TMC volume record data set (sorted)	Required
DSNB	DSNB data set (sorted)	Required
EDMNAME	EDM job name	Optional
VRSMGMT	VRS management value	Optional
STORE	Outcode location and storage location name	Optional (unless outcodes are used and will become RMM storage locations)
LOAN	Outcode location and loan location name	Optional (unless outcodes are used that are not already mapped to storage locations)
TMSPARM	The TMS PARMLIB options member	Optional
SGNAME	Volume ranges by scratch pool/Storage Group name	Optional

Note: If you do not use an optional input data set, you must specify DD DUMMY for it.

For TMC and DSNB records, first you must use the vendor-provided utility programs, TMSPTRS and TMSFVSN, to verify and ensure the integrity of the TMS database (for example, no broken volume or data set chains).

Do not rely on EDGC5LDR to fix the broken chains. The existence of broken chains may not be a problem during "dry runs" when you are only testing the conversion process to gain experience and do not plan to use the converted database in production mode.

You then use TMSDATA or, if TMSDATA is not available, DFSORT (see Figure 1 on page 11) to offload the TMC database to two data sets:

- Sequential data set containing the TMC volume records.
- Sequential data set containing the DSNB records.
Sort the sequential TMC volume data set and DSNB data set as follows (see Figure 3 on page 25).
- Sort the TMC volume records by the first VOLSER of the volume set, the volume sequence number, and the volume VOLSER in ascending order.
- Sort the DSNB records by the VOLSER on which file 2 starts and the file sequence number in ascending order.

Table 8. EDMNAME Identification Data Set Record Format

Column	Field description
1-8	EDM job name
9-16	Owner ID

Table 8. EDMNAME Identification Data Set Record Format (continued)

Column	Field description
17-21	EDM option (NOEDM or EDM). This option can be used to control data set recording. EDM Set data set recording OFF (Default) NOEDM Set data set recording ON
22-80	Comments

This figure shows a sample of how to identify EDM job names:

```
//EDMNAME DD *
DFHSM SYSOPER NOEDM job name - DFHSM, owner - SYSOPER not EDM
//* cols 1-8 EDM job name
//* 9-16 Default volume owner (optional)
//* 17-21 EDM option ( NOEDM or EDM ) 1
```

Figure 11. EDGC5LDR: EDM Identification Sample

The maximum number of EDM job names that can be identified through the EDMNAME DD data set is 20, as EDGC5LDR is originally coded. If this number is too small, you must alter the EDGC5LDR program to increase it.

Note: All EDM managed volumes will be converted to USER status, this indicates that this is a private volume which can be overwritten by any data set. If you do not identify EDM job names, the EDM volumes are converted as normal volumes. If any of these are multi-file volumes, but with only the first file recorded in TMS, you should convert them as EDM volumes or else DFSMSrmm will only allow file 1 to be used.

Table 9. VRSMGMT Management Value Data Set Record Format

Column	Field description
1-5	Expiration date
6-13	One- to eight-character VRS management value name
14-80	Comments

This figure shows a sample of how to create VRS management value names:

```
//VRSMGMT DD *
VRSMV=YES
99000D99000 While catalog retention
99001D99001 1 cycle/copy retention
99002D99002 2 cycles/copies retention
99003D99003 3 cycles/copies retention
98001D98001 1 day since last referenced
98002D98002 2 days since last referenced
98003D98003 3 days since last referenced
//* cols 1-5 Special meaning expiration date
//* 6-13 VRS management value
```

Figure 12. EDGC5LDR: VRS Management Value Name Sample

The maximum number of VRS management value names supported by EDGC5LDR is 1,000. If this number is too small, you must alter EDGC5LDR to increase it.

Note: The VRS management value names specified in the VRSMGMT DD data set are the names that your EDGUX100 exit should pass to RMM when it executes once you have RMM in record, warning, or protect mode.

If you plan to use any VRS management values, you must install the EDGUX100 exit.

If VRSMV=YES is not specified, only those keyword dates specified in VRSMGMT DD are converted to VRS management values. This enables you to move away from keyword dates.

If VRSMV=YES is specified, all keyword dates except 99365, are converted, and by default the VRS management value name is determined by the conversion programs, unless you also specify the keyword date and a management value to use. The conversion tools use some special VRS naming formats based on the keyword date value.

If you want a VRS management value assigned for 99365 you must code an entry in the VRSMGMT file. We suggest the use of the PERM VRS MV name because this is the value used by EDGCSRDS based on RDS input.

The names are similar to those used by TMS:

- catalog control EXPDT=99000, uses 'CATALOG'
- cycle control EXPDT=99ddd, uses 'CYCLEddd'
- last reference control EXPDT=98ddd, uses 'LDATEddd'
- catalog and days control EXPDT=90ddd, uses 'CATLGddd' with a NEXTVRS(CATALOG)

Any keyword dates specifically included in VRSMGMT can be defined with a VRS management value to use instead of building one using this default method.

All management values used are written as K-Records, and written to a new file VRSVALUE, which is input to EDGCSRDS processing.

Table 10. STORE Location Name Data Set Record Format

Column	Field description
1-4	Outcode location name
5-12	Default volume owner (optional)
13-20	Storage location name (optional)
21-80	Comments

This figure shows a sample of how to define outcode and storage location names:

```
//STORE DD *
ARC LOCAL
DIS ownerid DISTANT
/* You must use DD DUMMY if no data is supplied
/* cols 1 - 4 Outcode location name
/* 5 - 12 Default volume owner (optional)
/* 13 - 20 Storage location name (optional)
```

Figure 13. EDGC5LDR: STORE DDNAME Sample

EDGC5LDR is coded to a maximum number of 1,000 outcode location names to be translated to RMM's storage locations. If this number is too small, you must alter EDGC5LDR to increase it.

Use EDGC5BIN with the CHECK option to generate the list of vaults to convert as storage locations. You can customize the list as required.

Table 11. LOAN Location Name Data Set Record Format

Column	Field description
1-4	Outcode location name
5-12	Default volume owner (optional)
12-20	Loan location name (optional)
21-80	Comments

This figure shows a sample of how to define loan location names:

```
//LOAN    DD  *
BANK          Loan Location for Payroll Tapes
MFCH          Loan Location for Tapes sent to Microfiche Vendor
/* You must use DD DUMMY if no data is supplied
/* cols  1 - 4  Outcode location name
/*      5 - 12  Default volume owner (optional)
/*      12 - 20 Loan location name (optional)
```

Figure 14. EDGC5LDR: LOAN DDNAME Sample

You can have many RMM loan location names, and the names are not mandated by RMM. Therefore the outcode location names are used as RMM loan location names when converting outcodes to loan locations unless a loan location name is specified.

The maximum number of outcode location names that can be specified to be set to RMM loan locations by EDGC5LDR is 200. If this number is too small, you must alter EDGC5LDR to increase it.

Use EDGC5BIN with the CHECK option to generate the list of vaults to convert as loan locations. You can customize the list as required.

Table 12. SGNAME Pool Table Record Format

Column	Field description
1-6	Lowest volume serial number in the pool range
7-12	Highest volume serial number in the pool range
13-20	RMM pool name and volume storage group name
21-33	CA-1 SCRPOOL name

SGNAME file is created by EDGRCSR. This file lists tape ranges by pool name. Each record lists a single range and a pool name for EDGC5LDR to use to set storage group names for volumes.

Output Data Sets

EDGC5LDR creates output data sets containing the converted volumes and data set records, an error report file, and a warning report file

Table 13. EDGC5LDR Output Data Sets

DDNAME	Description
DEXTOUT	L- and D-Records data set The LRECL of this file is overridden during program execution. The value in the JCL examples and samples is just an indication of the approximate record length.
DEXTOUTK	K-Records data set The LRECL of this file is overridden during program execution. The value in the JCL examples and samples is just an indication of the approximate record length.
ERROR	Error report data set
REPORT	Warning report data set
VRSVALUE	Data set containing all created management values K-records This data set is input to EDGCSRDS processing.

Return codes

EDGC5LDR sets a return code when it ends:

Table 14. EDGC5LDR Return Codes

Return code	Description
0	Normal processing.
4	<ul style="list-style-type: none"> • A broken DSNB chain has been encountered and an attempt has been made to fix it. • A VRS management value name in the VRSMGMT DD data set begins with a blank character and is therefore unusable. • A Special Meaning TMC EXPDT was encountered that does not have a corresponding statement in the VRSMGMT DD data set. Permanent retention is forced in the Expiration Date field of the L-Record • A Special Meaning TMC EXPDT was encountered that does not conform to the format 98xxx or 99xxx. Permanent retention is forced in the Expiration Date Field of the L-Record. • A TMC outcode of blanks (x'40404040') was encountered: <ul style="list-style-type: none"> – If the slot assigned was zero, the current location in the L-Record is forced to SHELF. – If the slot assigned was nonzero, an error message is written stating that the outcode name was blank. The current location in the L-Record is forced to SHELF. • A TMC EXPDT has a format error.
8	There are extra DSNB records for a volume; that is, there are actually more DSNB records than indicated in the TMC volume record's DSNB count field.
12	Too many keyword dates were in use in the TMC. A table for 1000 maximum entries is used by EDGC5LDR. If more are used, the table size must be increased and EDGC5LDR re-assembled.
16	<ul style="list-style-type: none"> • A broken TMC volume chain has been encountered. • There are missing DSNB records; that is, fewer DSNB records were found than was indicated in the TMC volume record's DSNB count field. • A private (active) volume TMC record contains an invalid data set name. • The slot number assigned to one or more volumes that is being translated to storage location is zero. • A STORE DD control statement has no storage location name specified.

Messages

EDGC5LDR-00000 UNABLE TO PROCESS KEYWORD DATE yyddd - VRS TABLE IS FULL

Severity: Error

Explanation: Too many keyword dates were processed.

System action: Sets return code 12.

User response: Update the EDGC5LDR source code to use a value larger than 1,000 for the number of keyword dates supported, reassemble the updated code and rerun the job.

EDGC5LDR-00005 EXTRA DAYS AFTER CATALOG RETENTION VRS CREATED

Severity: Information

Explanation: The R9 VRS has been created based on the TMS R9 parmlib option.

User response: None.

EDGC5LDR-00010 EXTRA DAYS AFTER CYCLES RETENTION VRS CREATED

Severity: Information

Explanation: The RC VRS has been created based on the TMS RC parmlib option.

User response: None.

EDGC5LDR-00015 ABEND VRS CREATED

Severity: Information

Explanation: The ABEND VRS has been created based on the TMS ABE parmlib option.

User response: None.

EDGC5LDR-00020 EMPTY BIN COUNT GREATER THAN 99999999

Severity: Error

Explanation: You have requested to create more than 99999999 empty bins in this run. When CONSOLIDATE or REASSIGN is requested, you can specify how many empty bin records should be added to the storage locations.

System action: None.

User response: Specify a value between 1 and 99999999 and rerun the job.

Destination: PARMERR

EDGC5LDR-00025 VRS MGMT VALUE DATE ERROR: *input_card* - RC=4

Problem determination: where

input_card

INPUT CONVERSION CARD

Severity: Error

Explanation: EXPDT must be either 98xxx or 99xxx - check DDNAME(VRSMGMT).

System action: None.

User response: You can only specify a VRS management value either 98xxx or 99xxx. For the 90000 date you have to specify 99000, because the program will change the date to 90000 internally.

The maximum number of VRS management value names supported by EDGC5LDR is 1,000. Rerun the job.

Destination: VRSERR

EDGC5LDR-00030 VRS MGMT VALUE DATE ERROR: *input_card* - RC=4

Problem determination: where

input_card
INPUT CONVERSION CARD

Severity: Warning

Explanation: VRS management name must begin in column 6 - check DDNAME(VRSMGMT).

System action: None.

User response: Specified your one to eight character VRS management value name upon column 6 in the VRSMGMT DDNAME and rerun the job.

Destination: VRSERR2

EDGC5LDR-00035 NEW STORE NAME MISSING FOR CONSOLIDATE OPTION, CHECK DDNAME(STORE) - RC=16

Severity: Error

Explanation: Also check EXEC PARM if 'CONSOLIDATE' option is specified.

System action: None.

User response: Specify your new one to eight character storage location name upon column 13 in the STORE DDNAME and rerun the job.

Destination: STORERR

EDGC5LDR-00040 EXTRA DSNB RECORD FOUND, DSNBVOL(*vvvvvv*) AND DSNBFVOL(*ffffff*). RC = 8

Problem determination: where

vvvvvv Volume serial number of the extra DSNB record was found.

ffffff Volume serial number on which this extra DSNB record starts.

Severity: Error

Explanation: An additional DSNB record was found for volume *vvvvvv*.

System action: None.

User response: Check your CA-1 TMC and rerun the job.

Destination: XDSNBCK

EDGC5LDR-00045 *vvvvvv dddddd* -> MULTI-VOLUMES BROKEN CHAIN ERROR - RC = 16

Problem determination: where

vvvvvv Volume serial number.

dddddd Data set name.

Severity: Error

Explanation: Check if the volume being processed is the next volume in the multi-volumes chain.

System action: None.

User response: Correct your CA-1 TMC and rerun the job.

Destination: CHAINCK

EDGC5LDR-00050 PREVIOUS VOLSER (*pppppp*) AND NEXT VOLSER (*nnnnnn*) DSNB COUNT IS ZERO BUT DSNB RECORD FOUND

Problem determination: where

pppppp Previous volume serial number of data set.

nnnnnn Next volume serial number of data set.

Severity: Error

Explanation: Check if the volume being processed is the next volume in the multi-volumes chain and the number of data sets are greater than one.

System action: None.

User response: Correct your CA-1 TMC and rerun the job.

Destination: CHAINCK

EDGC5LDR-00055 EXTRA DSNB RECORD FOUND, DSNBVOL(*vvvvvv*) AND DSNBFVOL(*ffffff*) AND EDGC5LDR FIXED IT.

Problem determination: where

vvvvvv Volume serial number of the extra DSNB record was found.

ffffff Volume serial number on which this extra DSNB record starts.

Severity: Warning

Explanation: None.

System action: None.

User response: None.

Destination: XTRAERR

EDGC5LDR-00060 EXTRA DSNB RECORD FOUND, DSNBVOL(*vvvvvv*) AND DSNBFVOL(*ffffff*)

Problem determination: where

vvvvvv Volume serial number of the extra DSNB record was found.

ffffff Volume serial number on which this extra DSNB record starts.

Severity: Error

Explanation: The extra DSNB record was ignored.

System action: None.

User response: Correct your CA-1 TMC and rerun the job.

Destination: DSNXTRA

EDGC5LDR-00065 VOLSER WITH DSNBVOL(*vvvvvv*) AND DSNBFVOL (*dddddd*) HAS A BROKEN DSNB CHAINED, AND EDGC5LDR FIXED IT.

Problem determination: where

vvvvvv Volume serial number.

dddddd Volume serial number of the extra DSNB record was found.

Severity: Warning

Explanation: None.

System action: None.

User response: None.

Destination: CONTFIX

EDGC5LDR-00070 SLOT NUMBER 0000 WAS ASSIGNED TO VOLSER *vvvvvv* STORED IN *llll1* * LOCATION NOW SET TO *llll2*. MOVE VOLUME MANUALLY AFTER BY CMD

Problem determination: where

vvvvvv Volume serial number.

llll1 Storage location name.

llll2 Home location name.

Severity: Warning

Explanation: A volume assigned to a storage location was found, but the bin number (SLOT) is 0000.

System action: Location is set to home location.

User response: After conversion is done use the RMM TSO CHANGEVOLUME subcommand to set the correct location of the volume.

Destination: PUTSLOT

EDGC5LDR-00071 VIRTUAL VOLUME *vvvvvv* NOT LIBRARY RESIDENT, BUT NOT EXPORTED * CONVERTED AS NOT IN LIBRARY RC=4

Problem determination: where

vvvvvv Volume serial number.

Severity: Warning

Explanation: A virtual volume is not library resident and is not exported. It is converted as not in library.

System action: None.

User response: Correct your CA-1 TMC and rerun the job.

Destination: VTSCHK

EEDGC5LDR-00075 *oooo* STORE WAS NOT DEFINED IN STORE DD OR LOAN DD. VOLSER *vvvvvv* IS MARKED AS BEING STORED IN THE LIBRARY (CURRENT LOCATION IS BEING FORCED TO SHELF) RC=4

Problem determination: where

oooo Name of the out code or vault location.

vvvvvv Volume serial number.

Severity: Warning

Explanation: A volume assigned to an out of area location or a vault location was found, but the location name was either specified in the STOR DD-statement or the LOAN DD-statement.

System action: None.

User response: Add an additional information to the STORE DD-statement or to the LOAN DD-statement.

Destination: PUTLOCK

EDGC5LDR-00080 *oooo* STORE WAS NOT DEFINED IN STORE DD OR LOAN DD VOLSER *vvvvvv* WILL BE MARKED UNDER MANUAL MOVE CONTROL BY EDGCNVT - CURRENT LOCATION SET TO OUTCODE RC=4

Problem determination: where

oooo Name of the out code or vault location.

vvvvvv Volume serial number.

Severity: Warning

Explanation: An volume assigned to a out of area location or a vault location was found, but the location name was either specified in the STOR DD-statement or the LOAN DD-statement.

System action: None.

User response: Add an additional information to the STORE DD-statement or to the LOAN DD-statement. Or you can check the result after the conversion is finished and change the MOVEMODE to AUTO.

Destination: LIBMSG

EDGC5LDR-00085 VOLSER *vvvvvv* MAY BE AN EDM OWNED VOLUME, ITS CREATION JOB NAME IS *jjjjjjj* *
CONVERTED AS NON-EDM VOLUME - RC=4

Problem determination: where

vvvvvv Volume serial number.

jjjjjjj Creating job name.

Severity: Warning

Explanation: The out of area information was ignored.

System action: None.

User response: Correct your CA-1 TMC and rerun the job.

Destination: RESVOL

EDGC5LDR-00090 VOLUME *vvvvvv* NOT WITHIN VALID VOLUME RANGE RC=4

Problem determination: where

vvvvvv Volume serial number.

Severity: Warning

Explanation: The volume *vvvvvv* does not match any scratch pool entry supplied with SGNAME DD card.

System action: None.

User response: Correct your CA1-TMC and rerun the job.

Destination: INVNAME1

EEDGC5LDR-00095 VOLSER: *vvvvvv* DSN: *dsname* NO VRS MGMT VALUE SPECIFIED FOR DATE: *dddd*
CHECK DDNAME (VRSMGMT) THE EXPIRATION DATE HAS BEEN FORCED TO PERM RETENTION RC=4

Problem determination: where

vvvvvv Volume serial number.

dsname Data set name.

dddd VRS management value.

Severity: Warning

Explanation: A volser with a special meaning expiration date was found, but the expiration date was not specified in the VRSMGMT DD-statement.

System action: None.

User response: Add the missing VRS management value to the VRSMGMT DD-statement.

Destination: PUTEVRS

EDGC5LDR-00100 AN INVALID SPECIAL EXPIRATION DATE OF *xxxxxxx* HAS BEEN ENCOUNTERED FOR: *dddddd* VOLSER: *vvvvvv* DSN: *dsname* THE EXPIRATION DATE HAS BEEN FORCED TO PERM RETENTION RC=4

Problem determination: where

xxxxxxx Special meaning expiration date.

dddddd VRS management value.

vvvvvv Volume serial number.

dsname Data set name.

Severity: Warning

Explanation: You have specified a special meaning expiration date outside the range.

System action: None.

User response: You can only specify VRS management value names in the range 98ddd and 99ccc.

Destination: PUTCVMSG

EDGC5LDR-00105 NO VRS MGMT VALUE SPECIFIED FOR DATE: *dddddd* CHECK DDNAME (VRSMGMT) VOLSER: *vvvvvv* DSN: *dsname* THE EXPIRATION DATE HAS BEEN FORCED TO PERM RETENTION RC=4

Problem determination: where

dddddd VRS management value.

vvvvvv Volume serial number.

dsname Data set name.

Severity: Warning

Explanation: A DSNB record with a special meaning expiration date was found, but the expiration date was not specified in the VRSMGMT DD-statement.

System action: None.

User response: Add the missing VRS management value to the VRSMGMT DD-statement.

Destination: PUTVRSE

EDGC5LDR-00110 AN INVALID SPECIAL EXPIRATION DATE OF *xxxxxxx* HAS BEEN ENCOUNTERED FOR: *dddddd* VOLSER: *vvvvvv* DSN: *dsname* THE EXPIRATION DATE HAS BEEN FORCED TO PERM RETENTION RC=4

Problem determination: where

xxxxxxx Special meaning expiration date.

dddddd VRS management value.

vvvvvv Volume serial number.

dsname Data set name.

Severity: Warning

Explanation: You have specified a special meaning expiration date outside the range.

System action: None.

User response: You can only specify VRS management value names in the range 98ddd and 99ccc.

Destination: PUTCDMSG

EDGC5LDR-00115 * VOLSER *vvvvvv* MAY BE AN EDM OWNED VOLUME, ITS CREATION JOB NAME IS *jjjjjjj*. * DO NOT PROCEED WITH CONVERSION, PLEASE VERIFY AND CORRECT PROBLEM RC=16

Problem determination: where

vvvvvv Volume serial number.

jjjjjjj Creating job name.

Severity: Error

Explanation: Read the previous message.

System action: None.

User response: Correct the error in the CA-1 TMC and rerun the job.

Destination: PUTEND

EDGC5LDR-00120 * VOLSER *vvvvvv* CONTAINS INVALID DSNAME (IE. 1ST BYTE IS X'00'): *dsname* * RESDVOL+14 DSNBCJOB * VOLSER *vvvvvv* MAY BE AN EDM OWNED VOLUME, ITS CREATION JOB NAME IS *jjjjjjj*. * DO NOT PROCEED WITH CONVERSION, PLEASE VERIFY AND CORRECT PROBLEM RC=16

Problem determination: where

vvvvvv Volume serial number.

dsname Data set name.

jjjjjjj Creating job name.

Severity: Error

Explanation: An invalid data set name *dsname* on volume *vvvvvv* was found and the volume status was not scratch.

System action: None.

User response: Correct your CA1- TMC and rerun the job.

Destination: INVNAME2

EDGC5LDR-00125 *vvvvvv* PREVIOUS VOLSER *pppppp* (N/A) AND NEXT VOLSER *nnnnnn* (N/A) -> DSNB RECORD MISSING OR INCOMPLETE DSNB CHAINED RC=16

Problem determination: where

vvvvvv Volume serial number.

pppppp Volume serial number for the previous volume in the chain.

nnnnnn Volume serial number for the next volume in the chain.

Severity: Error

Explanation: The previous and next volume information not found.

System action: None.

User response: Correct the error in the CA-1 TMC and rerun the job. (You can use the TMSPTRS CA-1 utility to fix the problem).

Destination: DSNERR

EDGC5LDR-00130 BIN (SLOT) NUMBER WAS RESEQUENCED, STARTING WITH 000001* DUE TO 'REASSIGN' OR 'CONSOLIDATE' OPTION WAS SELECTED RC=4

Severity: Warning

Explanation: You have selected the REASSIGN or CONSOLIDATE execution parameter and the conversion program has changed some of the slot numbers.

System action: None.

User response: You must create empty bin records to provide space for RMM to move additional volumes to the storage locations after conversion.

Destination: CLOSE

EDGC5LDR-00135 * VOLSER *vvvvv* CONTAINS INVALID DSNNAME (IE. 1ST BYTE IS X'00'): *dsname* * VOLSER *vvvvv* MAY BE AN EDM OWNED VOLUME, ITS CREATION JOB NAME IS *jjjjjj*. * DO NOT PROCEED WITH CONVERSION, PLEASE VERIFY AND CORRECT PROBLEM RC=16

Problem determination: where

vvvvv Volume serial number.

dsname Data set name.

jjjjjj Creating job name.

Severity: Error

Explanation: An invalid data set name *dsname* on volume *vvvvv* was found and the volume status was not scratch.

System action: None.

User response: Correct your CA-1 TMC and rerun the job.

Destination: INVNAME1

EDGC5LDR-00140 CATALOG RETENTION VRS CREATED

Severity: Information

Explanation: 'While cataloged' next vital record specifications is defined to retain data set while it is cataloged.

System action: None.

User response: None.

Destination: VRSDAYS

Exported Logical Volumes and Stacked Volumes Support

EDGC5LDR detects exported logical volumes in the TMC data set and converts those as logical volumes. It also detects stacked volumes which are being used or have been used by export processing.

The data set name suffix 'T CONTAINR.VOLUME' is used to identify stacked volumes. It is best to use the EDGCNVT SYSIN file IF VOLRANGE with TYPE STACKED to define the complete range of stacked volumes. Stacked volumes are assumed to be library resident unless the outcode name is a vault.

To avoid problems with bin numbers for exported stacked volumes not matching the slot numbers you should avoid use of bin numbers in DFSMSrmm. Use the EDGCNVT SYSIN LOCDEF statement with NOBINS.

If you instead use BINS, when you run EDGUTIL with MEND(VOLUME) to enable stacked volume support the bin number from the first logical volume in sequence will be used instead of the existing assigned value from the stacked volume.

Creating the Empty Bin Records using EDGC5BIN

Note: The empty bin program, EDGC5BIN, should not be run, other than with the CHECK parameter, if the reassign or consolidate option is used in the execution of EDGC5LDR. The necessary empty bin records (E-Records) are provided by EDGC5LDR when the reassign or consolidate parameter is used.

RMM keeps track of all of the bins (slots) that exist in each of its shelf-managed storage locations. Each bin has a status of either "in-use" or "empty".

TMS/CA-1 only keeps track of the in-use slots. Therefore, the purpose of EDGC5BIN is to build E-Records that represent the empty RMM bins by recognizing gaps in the CA-1 in-use slot locations.

It also verifies the slot numbers and the outcode location names. If it encounters slot number 0 or multiple volumes in a single slot, it writes an error message to the ERROR DD data set.

Note: DFSMSrmm does not support a bin number of 0. The lowest bin number is 1. DFSMSrmm also does not allow more than one volume to be assigned to a given bin in an DFSMSrmm storage location.

If the outcode location name in the TMC volume record does not match one of the outcode location names specified in the STORE DD data set and the volume has a vault slot number assigned, EDGC5BIN writes an error message to the ERROR DD data set.

An execution parameter option is provided to allow the EDGC5BIN program to report all of the outcode locations and the in-use and empty slots in each outcode location.

See "Execution Parameter" on page 45 for more information.

Planning

Before executing EDGC5BIN, you must review the preprocessing considerations described in "Preparing to run the Extract" on page 10.

If you decide to reassign bin numbers and/or consolidate outcode locations through EDGC5LDR, do not execute EDGC5BIN without the CHECK parameter. EDGC5BIN requires that the records in the SLOTNO input file be sorted by outcode and slot number. This is achieved in the sort step.

The sort step is also used to include only those records required for each type of media; the volumes are selected as either tape reels or tape cartridges. When running with the CHECK parameter you can pass all volume records for processing. When creating the empty bin records, without the CHECK parameter, as long as all vaults are named in the STORE DD data set, you can also pass all volume records.

If you want to process only part of the vaults, use the SORT INCLUDE statement to select only the correct volume records by vault name.

If you have both tape reels and tape cartridges being sent to vault outcodes, CA-1 automatically subvaults the tape cartridges. DFSMSrmm can handle this using the MEDIANAME value, but you must run EDGC5BIN twice; once to process tape reels, and once to process cartridges. The sample EDGjxBIN does this.

You can separate the reels and cartridges at sort time by using the TMCDEN field. If you do not have both types of media you need only execute EDGC5BIN once.

Sample JCL

The figure below shows sample execution JCL for TMS/CA-1 Release 5.0 and above. Sample JCL is supplied in EDGJ5BIN.

```
/**
//DEL      EXEC PGM=IEFBR14
//DEXTOUT DD DISP=(MOD,DELETE),SPACE=(TRK,1),UNIT=SYSALLDA,
//          DSN=STSGWD.EDGCXBIN.DEXTOUT.DATA
//STOREOUT DD DISP=(MOD,DELETE),SPACE=(TRK,1),UNIT=SYSALLDA,
//          DSN=STSGWD.EDGCXBIN.STOREOUT.DATA
//LOANOUT  DD DISP=(MOD,DELETE),SPACE=(TRK,1),UNIT=SYSALLDA,
//          DSN=STSGWD.EDGCXBIN.LOANOUT.DATA
/**
/** SORT BY OUTCODE AND VAULT SLOT NUMBER
/**
//SORT      EXEC PGM=SORT,REGION=4M
//SYSOUT   DD SYSOUT=*
//SORTIN   DD DISP=SHR,DSN=STSGWD.TMS51.VOLRECS.DATA
//SORTOUT  DD DISP=(,PASS),AVGREC=K,SPACE=(340,(3,3),RLSE),
//          DSN=&&SLOT34,UNIT=SYSDA
//SYSIN    DD *
//          OPTION NOCHALT
//          SORT FIELDS=(109,4,CH,A,113,4,BI,A),DYNALLOC
/**
/** OPTION: PARM=CHECK
/** Use this option to quickly check what vaults are in use
/** and whether there are any potential problems.
/**EMPTYBIN EXEC PGM=EDGC5BIN,REGION=4M,COND=(0,NE),PARM='CHECK'
/**EMPTYBIN EXEC PGM=EDGC5BIN,REGION=4M,COND=(0,NE)
/**STEPLIB DD DISP=SHR,DSN=RMM.LOAD
//SYSUDUMP DD SYSOUT=*
//ERROR    DD SYSOUT=*
//SLOTNO   DD DISP=(OLD,DELETE),DSN=&&SLOT34
//DEXTOUT  DD DISP=(,CATLG),LRECL=512,RECFM=VB,DSORG=PS,
//          AVGREC=K,SPACE=(512,(1,1),RLSE),UNIT=SYSALLDA,
//          DSN=STSGWD.EDGCXBIN.DEXTOUT.DATA
/**STOREOUT DD DISP=(,CATLG),LRECL=80,RECFM=FB,DSORG=PS,
/**          AVGREC=K,SPACE=(80,(1,1),RLSE),UNIT=SYSALLDA,
/**          DSN=STSGWD.EDGCXBIN.STOREOUT.DATA
/**LOANOUT  DD DISP=(,CATLG),LRECL=80,RECFM=FB,DSORG=PS,
/**          AVGREC=K,SPACE=(80,(1,1),RLSE),UNIT=SYSALLDA,
/**          DSN=STSGWD.EDGCXBIN.LOANOUT.DATA
//STORE    DD DISP=SHR,DSN=STSGWD.EDGCXBIN.STOREOUT.DATA
/** cols 1- 4 Outcode location name
/**      5-12 Default volume owner (ignored)
/**     13-20 Storage location name (optional)
/**     21-26 Add empty bins option (optional) (ADDBIN or MAXBIN)
/**     27-32 Add empty bins number (optional) (mandatory when
/**           empty bins option is specified) (000000 - 999999)
/**           Must be 6 numeric digits. No leading or trailing blanks.
/**     33-80 Comments
```

Figure 15. EDGC5BIN: Sample JCL, CA-1 Release 5.0 and above

Execution Parameter

This describes the optional execution parameter of EDGC5BIN program:

CHECK

Use the CHECK parameter to identify all of your outcodes and produce input for the EDGC5LDR program. A pass is made through the TMC volume records to identify:

- All of the outcode locations in use and classify them as either store or loan type. See Table 17 on page 47 for the information that can be generated.

- Each occurrence of slot zero's use.
- Each occurrence of more than one volume in a given slot within a given outcode location.
- A summary of in-use and empty slot numbers for each outcode location.

Note: The STORE DD data set is not read when you use the CHECK parameter, which allows full analysis of all volume records, nor are E-Records produced.

Input Data Sets

EDGC5BIN has two input data sets:

Table 15. EDGC5BIN Input Data Sets

DDNAME	Description
SLOTNO	TMC volume records, representing all volumes, sorted by outcode location name and slot number in ascending order.
STORE	<p>Outcode location names with corresponding (optional) storage location names.</p> <p>The EDGC5BIN program does not use the Default Volume Owner field. Thus this data set can be the same data set that is used with EDGC5LDR. It should have the same data in it, if it is not the same data set.</p> <p>A maximum of 49 vaults can be specified.</p> <p>ADDBINxxxxxx or MAXBINxxxxxx option may be used to add additional E-Records when converting from TMS/CA-1 Release 5.0 and above.</p>

Table 16. STORE DD Input File Field Description

Column	Field description
1-4	Outcode location name
5-12	Default volume owner (ignored)
13-20	Storage location name (optional)
21-26	<p>Add empty bins option (optional)</p> <ul style="list-style-type: none"> • Valid values are ADDBIN or MAXBIN. • MAXBIN001000 means generate a total of 1000 bins. <p>Assume a store location has 500 in use bins and the highest in use bin number is 800 then 500 empty bin records will be created. 300 in the gaps between bin 1 and 800 and another 200 starting with bin number 801 until 1000.</p> <p>Note: If the highest in use bin number is equal or higher then the number in the MAXBIN parameter then no additional empty bin records will be created.</p> <ul style="list-style-type: none"> • ADDBIN001000 means generate 1000 empty bin records after the highest in use bin number. <p>Assume a store location has 500 in use bins and the highest in use bin number is 800 then starting with bin number 801 to 1800 another 1000 empty bin records will be created. This gives a total of 1800 bins.</p>
27-32	<p>Add empty bins number (optional) (mandatory when empty bins options is specified)</p> <ul style="list-style-type: none"> • Valid values 000000 - 999999. • Must be 6 numeric digits. • No leading or trailing blanks allowed.
33-80	Comments

To obtain the list of used vaults, for use as storage locations, you can run EDGC5BIN with the CHECK parameter.

This figure shows a sample of how to change outcode location names to storage location names:

```

/** You must supply the names of the Outcode Location names
/** that will become Storage Locations when RMM is implemented.
//STORE DD *
SAFE LOCAL <--- SAFE Outcode to be translated to RMM's LOCAL
ARC DISTANT <--- Storage Location, ARC to DISTANT, and VLT1
VLT1 REMOTE <--- to REMOTE.
LOCA RMMLOC MAXBIN001000 <--- Have at least 1000 bins in RMMLOC
LOCB RMMLOC1 ADDBIN001000 <--- Add 1000 Empty bins to RMMLOC1
/** cols 1-4 Outcode location name
/** 5-12 Default volume owner (ignored)
/** 13-20 Storage location name (optional)
/** 21-26 Add empty bins option (optional) (ADDBIN or MAXBIN)
/** 27-32 Add empty bins number (optional) (mandatory when
/** empty bins option is specified)
/** Must be 6 numeric digits. No leading or trailing blanks
/** 33-80 Comments

```

Figure 16. EDGC5BIN: STORE DDNAME Sample

Output Data Sets

EDGC5BIN creates an output data set containing E-Records that represent the empty bin in each of the outcode locations and an error report data set.

When you use the CHECK parameter, a list of all outcodes by classification can be produced in the LOANOUT DD and STOREOUT DD data sets.

Table 17. EDGC5BIN Output Data Sets

DDNAME	Description
DEXTOUT	E-Record data sets. The LRECL of this file is overridden during program execution. The value in the JCL examples and samples is just an indication of the approximate record length.
ERROR	Error messages.
STOREOUT	A list of all outcodes using slot numbers. This should match the vaults defined to VMS. Use the file as input to EDGC5LDR STORE DD. This file is optional and is used only if CHECK is requested.
LOANOUT	A list of all outcodes that do not use slot numbers. This should match the codes you use outside VMS. Use the file as input to the EDGC5LDR LOAN DD. This file is optional and is used only if CHECK is requested.

Return Codes

EDGC5BIN sets a return code when it terminates:

Table 18. EDGC5BIN Return Codes

Return code	Description
0	Normal processing.
4	<ul style="list-style-type: none">• Outcode location name from the TMC volume record did not match any of the outcode location names specified in the outcode location name fields in the STORE or LOAN DD data sets.• Invalid additional bins option.• Invalid additional bins number.
16	<ul style="list-style-type: none">• Slot number 0 is in use.• More than one volume is assigned a single slot number.
20	Both return code 4 and 16 problems were found.

Messages

EDGC5BIN-00000 *tmcvol tmcdsn tmconame* * HAS AN INVALID VAULT NAME (OUTCODE)

Problem determination: where

tmcvol Input TMC record volser.

tmcdsn Input TMC REDATA set name.

tmconame
Input TMC record storage location.

Severity: Warning

Explanation: None.

System action: The job continues and sets a minimum return code of 4.

User response: None.

EDGC5BIN-00005 *tmcvol tmcdsn tmconame* * BIN NUMBER 0 IS NOT VALID WITH RMM

Problem determination: where

tmcvol Input TMC record volser.

tmcdsn Input TMC REDATA set name.

tmconame
Input TMC record storage location.

Severity: Error

Explanation: None.

System action: The job continues and sets a minimum return code of 16.

User response: None.

EDGC5BIN-00010 *tmcvol tmcdsn tmconame* * ASSIGNED TO BIN # *tmcslot* WHICH WAS THE SAME AS BIN # ASSIGNED TO VOLSER

Problem determination: where

tmcvol Input TMC record volser.

tmcdsn Input TMC REDATA set name.

tmconame
Input TMC record storage location.

tmcslot Input TMC record slot number.

Severity: Error

Explanation: None.

System action: The job continues and sets a minimum return code of 16.

User response: None.

EDGC5BIN-00015 *ssss* **HAS** *uuuuuu* **IN-USE BIN NUMBER AND** *eee* **EMPTY BIN NUMBER**

Problem determination: where

ssss DFSMSrmm storage location name.

uuuuuu Number of used bins for this storage location.

eee Number of empty bins for this storage location.

Severity: Information

Explanation: This message tells you the number of bins for this storage location.

System action: None.

User response: None.

EDGC5BIN-00020 *ssss* **HIGHEST IN-USE BIN NUMBER IS** *uuuuuuu* **MEDIANAME** *mmmmmmmmmm*

Problem determination: where

ssss DFSMSrmm storage location name.

uuuuuuu
Number of used bins for this storage location.

mmmmmmmmmm
Media name for this storage location.

Severity: Information

Explanation: None.

System action: None.

User response: None.

EDGC5BIN-00025 *ssss* *eeeeee* **ADDITIONAL EMPTY BINS CREATED**

Problem determination: where

ssss DFSMSrmm storage location name.

eeeeee Number of empty bins.

Severity: Information

Explanation: None.

System action: None.

User response: None.

EDGC5BIN-00030 ssss INVALID VALUE eeeee FOR NUMBER OF ADDITIONAL EMPTY BINS

Problem determination: where

sss DFSMSrmm storage location name.

eeeeee Number of empty bins.

Severity: Error

Explanation: None.

System action: The job continues and sets a minimum return code of 16.

User response: None.

EDGC5BIN-00035 ssss INVALID VALUE 00000 FOR ADDITIONAL EMPTY BINS OPTION

Problem determination: where

sss DFSMSrmm storage location name.

00000 Used value. Can only be one of MAXBIN or ADDBIN.

Severity: Error

Explanation: None.

System action: The job continues and sets a minimum return code of 16.

User response: None.

EDGC5BIN-00040 ssss BIN NUMBER LIMIT 999999 HAS BEEN REACHED. TOTAL NUMBER OF BINS ADAPTED TO 999999

Problem determination: where

sss DFSMSrmm storage location name.

Severity: Warning

Explanation: None.

System action: The job continues and sets a minimum return code of 16.

User response: None.

Overview of Policy Conversion

Conversion of retention and movement policies consists of multiple parts:

1. Conversion of TMC expiration dates, which may be real dates or keyword dates.
2. Determining the CA-1 parmlib options that affect policies, so that they can be built-in to the DFSMSrmm policies conversion creates.
3. Conversion of the RDS entries to both policies and values for use in the EDGUX100 installation exit and SMS ACS statements.
4. Conversion of the VPD entries to DFSMSrmm VRS policies.
5. Conversion of the special CA-1 pattern mask character to the equivalent DFSMSrmm mask character.

DFSMSrmm policy management provides the support required for this. EDGHSKP VRSEL processing supports up to two policies per data set:

- for identifying retention
- for identifying movement/vaulting

This is particularly important when keyword dates are used to identify retention and VPD entries also exist for the same data sets.

Conversion proceeds as follows:

1. TMC expiration dates which are keyword dates, are used to generate a DFSMSrmm VRS management value. The VRS management value is assigned to the data set, and the expiration date is set to the assign/creation date.

A K record is created for each generated VRS management value. EDGC5LDR handles this automatically when the VRSMV=YES option is coded in the VRSMGMT file.

2. Each of the programs, EDGC5LDR, EDGCSRDS, and EDGCSVDS obtain the CA-1 parmlib options that affect policies. They build them into the DFSMSrmm policies they create.
3. EDGCSRDS converts the RDS entries to DFSMSrmm VRS policies (K-Records at this stage) using the data set name in the VRS as a VRS management value. Only keyword dates and special CA-1 retention options are converted to management values. Any RETPD retention values are not converted to VRSs.

EDGCSRDS also converts each of the RDS entries, together with the VRSVALUE file input from EDGC5LDR, into a table for use with the EDGUX100 installation exit. The table, UXTABLE, is an assembler source code table which you can later customize and modify independently of the EDGUX100 installation exit.

A better alternative than using the UXTABLE with EDGUX100 is to use the RMM SMSACS parmlib option and the ACS statements created by the EDGRACS exec.

4. EDGRACS reads the UXTABLE from EDGCSRDS and creates a data set designated by DD EDGACS with ACS statements and a data set designated by DD EDGCMD with RMM ADDVRS commands. Instead of using the UXTABLE you can merge the generated ACS statements with existing ACS routines and run the created RMM ADDVRS subcommands after creation of the CDS.
5. EDGCSVDS converts the VPD entries to K-Records which become DFSMSrmm VRS policies later in the conversion process.

The K-Records will be equivalent to the rules defined in the CA-1 VPD.

EDGCNVT converts the K-Records to DFSMSrmm VRS policies, and when you run EDGHSKP with the VRSEL execution parameter, DFSMSrmm determines which policies to use for each data set and updates the data set information in the DFSMSrmm CDS with the VRS details it determines, such as:

- retention date,
- matching primary and secondary VRS,
- and whether the data set is VRS retained.

For correct and complete policy conversion you must:

- Run each of the 4 programs EDGC5LDR, EDGCSRDS, EDGRACS and EDGCSVDS, using the sample.
- Convert the K-Records using EDGCNVT.
- Decide whether you want to manage your retention policies by using UXTABLE or by implementing it into your SMS ACS routines.

The recommendation is to use the ACS statements with the RMM SMSACS parmlib option instead of the UXTABLE, and to use the sample EDGUX100 exit.

- If UXTABLE will not be used, you have to merge the generated ACS statements with existing ACS routines, define the new management classes, activate the ACS routines and run the created RMM ADDVRS subcommands.
- If UXTABLE will be used, ensure that UXTABLE is loaded at IPL time for use by EDGUX100. Use the sample EDGCVRSP procedure to do this.

Creating VRS Records using EDGCSRDS

The EDGCSRDS program converts the CA-1 retention policy definitions from the Retention Data Set (RDS) to K-Records and a data set designated by DD VRSTABLE that can be used for building SMS ACS statements or as the source for a dynamic table for the EDGUX100 installation exit.

The created K-Records are "Data Set Name Vital Record Specifications" types, but use a single data set name qualifier that identifies the VRS management value.

Also the EDGCSRDS program will analyze the CA-1 parmlib member (CAL.PPOPTION) to convert the:

- RC** Extended CYCLE retention
- R9** Extended CATALOG retention
- RP** Default retention
- CYD** Cycle control option
- RO** Retention override option
- CJOB** Cycle control by jobname

information in DFSMSrmm CDS K-Records.

No reference is made to the Vault Pattern Description Data Set (VPD), so no matching information is provided or required.

How the Conversion Process Works

The Retention Data Set (RDS) input is converted to values to be used to create SMS ACS statements and RMM ADDVRS statements by the EDGRACS sample or by a modified EDGUX100 installation exit sample to assign retention at OPEN time.

This includes an entry as a default for all data set using the RP management value. If RO YES is specified, the converted values will be used to replace any JCL retention values, including keyword dates.

The following figure shows the table used by the EDGUX100 installation exit:

* start of RDS entries		
EDGCVRSR DSN=A.B*,		X
JOBNAME=J1,		X
RO=YES		X
VRSVAL=LDATE003		
EDGCVRSR DSN=BACKUP.DATA.G%%V%,		X
JOBNAME=*,		X
RO=YES		X
VRSVAL=CYCLE002		
EDGCVRSR DSN=BACKUP.DATA,		X
JOBNAME=*,		X
RO=YES		X
VRSVAL=CYCLE002		
EDGCVRSR DSN=OFFSITE*,		X
JOBNAME=*,		X
RO=YES		X
VRSVAL=CATLG005		
* default RP value		
EDGCVRSR DSN=*,	default RP value	X
RO=NO,		X
VRSVAL=CATALOG		
* start of keyword dates from EDGC5LDR		
EDGCVRSR KEYDATE=99000,		X
VRSVAL=WHILECAT		
EDGCVRSR KEYDATE=99005,		X
VRSVAL=D99005		

Figure 17. Example of Data Produced from the RDS

The table is an input for the EDGRACS exec in the EDGJSRDS sample job that translates the VRS management values generating ACS statements and RMM ADDVRS subcommands. The table is also assembled by the EDGJSRDS sample job into a separate load module called UXTABLE, that can be changed dynamically using the EDGCVRSL program from the EDGCVRSP procedure.

If the RP keyword is specified in the CA-1 parmlib member, a table entry for DSN=* is created and either the RETPD=ddd is coded or the VRSVAL=keydate is coded depending on whether the default is a retention period or a keyword date.

The keyword date management value uses the same naming convention as described earlier.

The last K-records in a chain generated for catalog or cycles retention are built including a NEXTVRS(RC) or NEXTVRS(R9). The K-records for RC and R9 that specify EXTRADAYS COUNT(nnn) are assumed to be generated by the EDGC5LDR conversion program.

RC is extra days after cycles retention, and R9 is extra days after catalog retention. If the RO=YES option is specified in the CA-1 parmlib member, each management value K-record generated includes the RELEASE(EXPIRYDATEIGNORE) option.

Any cycle control K-records use BYDAYSCYCLE as determined by the CA-1 parmlib option CDAY=DAY, and include JOBNAME(*) as determined by CJOB if CJOB=JOB is specified.

The next figure shows examples of Vital Record Specifications (VRS) built by the EDGCSRDS conversion program:

```

RMM ADDVRS DSN('LDATE003') -
  LASTREFERENCE -
  COUNT(3) -

RMM ADDVRS DSN('CYCLE002') -
  CYCLES -
  COUNT(2) -
  NEXTVRS(RC)

RMM ADDVRS DSN('CATLG005') -
  DAYS -
  COUNT(5) -
  NEXTVRS(CATALOG)

RMM ADDVRS NAME(CATALOG) -
  WHILECATALOG -
  NEXTVRS(R9)

RMM ADDVRS DSN('CATALOG') -
  WHILECATALOG -
  NEXTVRS(R9)

```

Figure 18. The VRS built by EDGCSRDS

Sample JCL

This figure shows the sample execution JCL. See sample EDGJSRDS for the supplied sample JCL.

```

// SET RMMLIB=xxxxxxx
//*
//DEL EXEC PGM=IEFBR14
//DEXTOUT DD DISP=(MOD,DELETE),SPACE=(TRK,1),UNIT=SYSALLDA,
// DSN=STSGWD.EDGCSRDS.DEXTOUT.DATA
//VRSTABLE DD DISP=(MOD,DELETE),SPACE=(TRK,1),UNIT=SYSALLDA,
// DSN=STSGWD.EDGCSRDS.VRSTABLE.DATA
//EDGACS DD DISP=(MOD,DELETE),SPACE=(TRK,1),UNIT=SYSALLDA
// DSN=STSGWD.EDGRACS.EDGACS.DATA
//EDGCMD DD DISP=(MOD,DELETE),SPACE=(TRK,1),UNIT=SYSALLDA
// DSN=STSGWD.EDGRACS.EDGCMD.DATA
//* REFORMAT THE RDS RECORDS FOR INPUT TO SORT
//EDGCRFMT EXEC PGM=EDGCRFMT
//*STEPLIB DD DISP=SHR,DSN=RMM.LOAD
//IFILE DD DISP=SHR,DSN=STSGWD.TMS.RDS.DATA
//OFFILE DD DISP=(,PASS),DSN=&&RFMTRDS,UNIT=SYSALLDA,
// AVGREC=K,SPACE=(80,(1,1),RLSE)
//*
//EDGCSRDS EXEC PGM=EDGCSRDS,REGION=4M,PARM='BOTH'
//*STEPLIB DD DISP=SHR,DSN=RMM.LOAD
//SYSUDUMP DD SYSOUT=*
//ERROR DD SYSOUT=*
//PERMRPT DD SYSOUT=*
//RDS DD DISP=(MOD,PASS),DSN=&&SORTRDS
//VRSVALUE DD DISP=SHR,DSN=STSGWD.EDGCXLDV.VRSVALUE
//TMSPARM DD DISP=SHR,DSN=STSGWD.TMS50.PARMLIB.DATA
//DEXTOUT DD DISP=(,CATLG),LRECL=512,RECFM=VB,DSORG=PS,
// AVGREC=K,SPACE=(512,(2,2),RLSE),UNIT=SYSALLDA,
// DSN=STSGWD.EDGCSRDS.DEXTOUT.DATA
//VRSTABLE DD DISP=(,CATLG),AVGREC=K,SPACE=(80,(100,100),RLSE),
// DSN=STSGWD.EDGCSRDS.VRSTABLE.DATA,UNIT=SYSALLDA
//*
//EDGRACS EXEC PGM=IKJEFT01,PARM='%EDGRACS',DYNAMNBR=20
//SYSPROC DD DISP=SHR,DSN=&RMMLIB
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD DUMMY
//VRSTABLE DD DISP=SHR,DSN=&RMMLIB(EDGCVRSF)
// DD DISP=SHR,DSN=STSGWD.EDGCSRDS.VRSTABLE.DATA
// DD DISP=SHR,DSN=&RMMLIB(EDGCVRSE)
//EDGACS DD DISP=(NEW,CATLG),DSN=STSGWD.EDGRACS.EDGACS.DATA,
// UNIT=SYSALLDA,SPACE=(TRK,(1,1)),LRECL=80,RECFM=FB
//EDGCMD DD DISP=(NEW,CATLG),DSN=STSGWD.EDGRACS.EDGCMD.DATA,
// UNIT=SYSALLDA,SPACE=(TRK,(1,1)),LRECL=80,RECFM=FB
//*
```

Figure 19. EDGCSRDS: Sample JCL (Part 1 of 2)

```

//*
//* FOLLOWING STEPS ASSEMBLES AND LINKAGE EDITS THE UXTABLE.
//* THE UXTABLE CONSISTS OF 3 PARTS:
//* 1. EDGCVRSF (FROM RMMLIB)
//* 2. THE ABOVE CREATED VRSTABLE
//* 3. EDGCVRSE (FROM RMMLIB)
//*
// EXEC HLASM
//C.SYSIN DD DISP=SHR,DSN=&RMMLIB(EDGCVRSF)
//          DD DISP=SHR,DSN=STSGWD.EDGCSRDS.VRSTABLE.DATA
//          DD DISP=SHR,DSN=&RMMLIB(EDGCVRSE)
//C.SYSLIB DD DISP=SHR,DSN=SYS1.MACLIB
//          DD DISP=SHR,DSN=SYS1.MODGEN
//          DD DISP=SHR,DSN=&RMMLIB
//LINKEDIT EXEC PGM=IEWL,PARM='LIST'
//SYSLMOD DD DSN=SYS1.RMM.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSLIN DD DSN=&&OBJ,DISP=(OLD,DELETE)
//          DD *
//          NAME UXTABLE(R)
//*

```

Figure 19. EDGCSRDS: Sample JCL (Part 2 of 2)

Execution Parameters

The data set names specified in the RDS and VPDD can be used to match to both standard data set names and generations of a GDG where the base name matches the data set mask.

This processing by TMS is independent of whether a fully qualified name or an index is specified. In DFSMSrmm, if a data set name mask is specified fully qualified, it can only match to a standard data set name unless the GDG operand is also specified for the VRS.

If the GDG operand is specified, standard data set names can no longer match to the mask.

In DFSMSrmm you can get the same results in these ways:

- Specify a generic data set name mask, so that both GDGs and standard data set names can match:
ADDVRS DSN('A.B.*') ...
- Create a VRS specifying the specific data set name without the GDG operand and create a second mask using a pseudo-GDG mask to manage the GDG generations:
ADDVRS DSN('A.B') NOGDG ...
ADDVRS DSN('A.B.G^^^V^^') NOGDG ...

The extract programs use the latter method to most specifically match the retention and movement policies of TMS. However, you can control what EDGCSRDS does by specifying one of the execution parameters--GDG, NONGDG, or BOTH.

When BOTH is specified for EDGCSRDS the UXTABLE table entries are created using
DSN=A.B.G%%%%V%%.

EDGCSRDS optionally requires an execution parameter.

This describes the execution parameter of EDGCSRDS program:

GDG Specify this parameter if the fully qualified data set names in the RDS are only used to match the GDG base names. If any of the data set names match the standard data sets, use the BOTH parameter.

NONGDG

(Default) Specify this parameter if the fully qualified data set names in the RDS never match a GDG base name. If any of the data set names match GDG base names, use the BOTH parameter.

BOTH Specify this parameter if the fully qualified data set names in the RDS can potentially match either standard data set names or GDG base names. If you know more specifically what they match, consider either the GDG or NONGDG parameter.

Input Data Sets

EDGCSRDS has three input data sets:

Table 19. EDGCSRDS Input Data Sets

DDNAME	Description
VRSVALUE	Data set containing all created management values K-records. This data set is created by the EDGC5LDR processing.
RDS	The RDS sorted by data set and job name in descending order (Required).
TMSPARM	The TMS PARMLIB options member (Optional).

Note: If you do not use an optional input data set, you must specify DD DUMMY for it.

Output Data Sets

EDGCSRDS creates output data sets and error reports:

Table 20. EDGCSRDS Output Data Sets

DDNAME	Description
DEXTOUT	K-Records The LRECL of this file is overridden during program execution. The value in the JCL examples and samples is just an indication of the approximate record length.
ERROR	Information and error messages.
PERMRPT	Permanent Retention Report.
VRSTABLE	All the new EDGUX100 dynamic table entries are written to this file. They are later merged with samples EDGCVRSE and EDGCVRSE and assembled into the UXTABLE load module.

Return Codes

EDGCSRDS sets the return code accordingly when it ends:

Table 21. EDGCSRDS Return Codes

Return code	Description
0	Normal processing.
4	Permanent retention policy found but K-Record is created.
16	Unable to create K-Records because of RDS or VPDD control statement error.

Messages

EDGCSRDS-00000 PERMANENT RETENTION POLICY FOUND - *ccard* - RC=04 CHECK ERROR FILE FOR ADDITIONAL MESSAGES IF ANY IF NO MORE MESSAGES, K RECORD IS WRITTEN TO DEXTOUT DD WITH VITAL COUNT SET TO 99,999

Problem determination: where

ccard VRSVALUE input data set record.

Severity: Information

Explanation: None.

System action: The job continues and sets a minimum return code of 4.

User response: None.

EDGCSRDS-00010 ERROR *dddddddddd* IS AN UNSUPPORTED DATE FORMAT

Problem determination: where

dddddddddd
CA-1 PARMLIB options member tms date format.

Severity: Error

Explanation: None.

System action: The job terminates with return code 16.

User response: Correct the error in the CA-1 PPOPTION member and rerun the job.

EDGCSRDS-00015 RDS CONTROL CARD ERROR: *ccard* - RC=16

Problem determination: where

ccard RDS input data set record.

Severity: Error

Explanation: The error can be caused by the date format not matching that in the options file or used by default.

System action: The job terminates with return code 16.

User response: Correct the error and rerun the job.

Creating SMS ACS Statements using EDGRACS

The EDGRACS exec translates the VRS management values from the VRSTABLE created by EDGCSRDS, generating ACS statements and placing them into the data set designated by DD name EDGACS, and also generating RMM ADDVRS commands and placing them in the data set designated by DD name EDGCMD.

EDGACS contains ACS FILTLIST and WHILE-SET statements. You can update the statements based on your required policies and merge them with the existing ACS routines.

EDGCMD contains RMM ADDVRS commands, for RETPD entries from VRSTABLE, that updates the RMM CDS. Before running the commands you can manually update the ADDVRS statements according to your required policies.

EDGRACS is called by the EDGJSRDS sample JCL. (See Figure 19 on page 54).

EDGRACS scans the UXTABLE and eliminates all the ones with duplicate keydate or a data set and job name combination and issues informational messages.

The SMS ACS statements are divided into the following groups:

GROUP 1

Contains the statements that can override any specified JCL RETPD or EXPDT value. Any matching data set has the expiration date and management class set using the current table entry values.

GROUP 2

Contains the statements for keydate VRS management.

GROUP 3

Contains the statements that accept the JCL RETPD or EXPDT value if they are set.

This could result in changing the selection order of the retention policy comparing to the UXTABLE, because the order now depends on the order of the WHEN statements.

Input Data Sets

EDGRACS has three input data sets:

Table 22. EDGRACS Input Data Sets

DDNAME	Description
VRSTABLE	<ul style="list-style-type: none">• EDGCVRSF Beginning of the VRSTABLE. With &DFRETPD a default retention period can be set.• VRSTABLE A table that, among other information, contains retention policies to be assigned at OPEN time.• EDGCVRSE Ending of the VRSTABLE. With default DSN='*' set.

Output Data Sets

EDGRACS creates output data sets:

Table 23. EDGRACS Output Data Sets

DDNAME	Description
EDGACS	<p>A data set containing ACS statements and filter lists for management class processing.</p> <p>It also contains a note with a recommendation of further EDGUX100 preparation depending on the results of the translation.</p>
EDGCMD	<p>A data set containing RMM ADDVRS commands for RETPD entries from VRSTABLE.</p>

Return Codes

EDGRACS sets the return code accordingly when it ends:

Table 24. EDGRACS Return Codes

Return code	Description
0	Normal processing.
4	Warning message has been issued. Duplicate records found. See joblog.

Table 24. EDGRACS Return Codes (continued)

Return code	Description
8	Error message has been issued. Wrong input data. See joblog.
12	Input or Output error when reading or writing a file.

Messages

EDGRACS-00005 I/O ERROR WHEN *action* THE FILE *file*, RC=*rc*

Problem determination: where

action READING, WRITING, READING/CLOSING, WRITING/CLOSING.

file DD NAME. can be VRSTABE, EDGACS, or EDGCMD.

rc EXECIO return code. See OS/390 TSO REXX User's Guide.

Severity: Error

Explanation: An error during reading or writing a specified file.

System action: The current input request fails.

User response: Check, correct, and rerun your job.

EDGRACS-00010 DUPLICATE KEYDATE = *keydate*. ONLY THE FIRST RECORD IS GOING TO BE PROCESSED

Problem determination: where

keydate EDGVRSG macro keyword in the file VRSTABLE.

Severity: Warning

Explanation: A duplicate keydate is found. The first record in the VRSTABLE is going to be processed. All other occurrences are going to be ignored.

System action: EDGRACS continues processing.

User response: Check the VRSTABLE file, correct it and rerun your job.

EDGRACS-00015 DUPLICATE DSN=*dsn*,JOB=*job*. ONLY THE FIRST RECORD IS GOING TO BE PROCESSED

Problem determination: where

dsn,job EDGVRSG macro keywords in the file VRSTABLE.

Severity: Warning

Explanation: A duplicate DSN, JOB is found. The first record in the VRSTABLE is going to be processed. All other occurrences are going to be ignored.

System action: EDGRACS continues processing.

User response: Check the VRSTABLE file, correct it and rerun your job.

EDGRACS-00020 INVALID EDGCVRSRSG MACRO KEYWORD: *keyword*

Problem determination: where

keyword EDGVRSG macro keyword in the file VRSTABLE.

Severity: Warning

Explanation: An invalid keyword has been encountered in the VRSTABLE file. The following keywords are defined: DSN, JOB, VRSVAL, POOL, DSECT, RETPD, ACLOPT, KEYDATE, RO, PGM, and F1ONLY.

System action: EDGRACS sets RC 8 and continues processing.

User response: Check the VRSTABLE file, correct it and rerun your job.

Creating VRS Records using EDGCSVDS

EDGCSVDS converts the vaulting policies from the VPDD into K-Records representing the RMM storage location movement policies. DFSMSrmm does not require that the retention type of all locations in a multiple locations movement policy be the same.

The EDGCSVDS program converts the CA-1 vault pattern description (VPDD) options by the following rules:

OR The EDGCSVDS program detects multiple options on a single V= statement and converts them using ANDVRS. This matches the CA-1 'OR' condition.

AND Multiple options on separate V= statements are converted using NEXTVRS. This matches the CA-1 'AND' condition.

VR= In other than the first vault is converted to EXTRADAYS in a NAME vital record specification (VRS). VR= in the first vault is still converted as DAYS (CR=) in a DSNAME vital record specification (VRS).

In cases where VR= is combined in an "OR" case with another retention type, such as:

```
DSN=dsname.-,SEPDSN
    V=VLT2,VR=42,C=1
    V=VLT4,VR=42,C=1
```

there is no direct equivalent in RMM because of the restrictions on the use of EXTRADAYS with ANDVRS. By altering the second VR=42, to CR=84 conversion can be successful and give you similar results.

```
DSN=dsname.-,SEPDSN
    V=VLT2,VR=42,C=1
    V=VLT4,CR=84,C=1
```

LU= Is converted to LAST REFERENCE DAYS.

CJOB=

Any cycle control K-records use BYDAYSCYCLES as determined by CYD=YES and include JOBNAME(*) as determined by the VPD entry.

MASKS

Generic data set name masks are converted to the nearest DFSMSrmm equivalent. See Table 28 on page 64 for examples.

Note: All K-records created include retention type

- BYDAYSCYCLES
- CYCLES
- DAYS
- LASTREFERENCEDAYS
- EXTRADAYS
- UNTILEXPIRED
- WHILECATALOG

and count value COUNT(days/cycles).

If you decide to reassign the storage location names when running EDGC5LDR, EDGCSVDS needs to know the new name of each of the storage locations; see Table 25 on page 62 for more information.

The EXP value is converted to UNTILEXPIRED, and should work the same way as it did under TMS. If a data set matches to a data set name VRS, converted from the VPD, with UNTILEXPIRED, and also has a VRS management value assigned, the VRS management value determines how long the data set name VRS retains the data set.

For data sets that have no VRS management value the volume expiration date is used by DFSMSrmm.

Sample JCL

The following figure shows the sample execution JCL. See sample EDGJSVDS for the supplied sample JCL.

```

/**
//DEL      EXEC PGM=IEFBR14
//DEXTOUT DD DISP=(MOD,DELETE),SPACE=(TRK,1),UNIT=SYSALLDA,
//          DSN=STSGWD.EDGCSVDS.DEXTOUT.DATA
/**
//EDGCSVDS EXEC PGM=EDGCSVDS,PARM='BOTH'
/**STEPLIB DD DISP=SHR,DSN=RMM.LOAD
//SYSUDUMP DD SYSOUT=*
//ERROR    DD SYSOUT=*
//EXPRPT   DD SYSOUT=*
//STORE    DD DISP=SHR,DSN=STSGWD.EDGCXBIN.STOREOUT.DATA
//VDS      DD DISP=SHR,DSN=STSGWD.TMS.VPDD.DATA
//DEXTOUT DD DISP=(,CATLG),UNIT=SYSALLDA,LRECL=512,
//          RECFM=VB,SPACE=(500,(200,100),RLSE),
//          DSN=STSGWD.EDGCSVDS.DEXTOUT.DATA
//TMSPARAM DD DISP=SHR,DSN=STSGWD.TMS50.PARMLIB.DATA

```

Figure 20. EDGCSVDS: Sample JCL

Execution Parameter

The data set names specified in the RDS and VPDD can be used to match to both standard data set names and generations of a GDG where the base name matches the data set mask. This processing by CA-1 is independent of whether a fully qualified name or an index is specified.

In DFSMSrmm, if a data set name mask is specified fully qualified, it can only match to a standard data set name unless the GDG operand is also specified for the VRS. If the GDG operand is specified, standard data set names can no longer match to the mask.

In DFSMSrmm you can get the same results in these ways:

- Specify a generic data set name mask, so that both GDGs and standard data set names can match:
ADDVRS DSN('A.B.*') ...
- Create a VRS specifying the specific data set name without the GDG operand and create a second mask using a pseudo-GDG mask to manage the GDG generations:
ADDVRS DSN('A.B') NOGDG ...
ADDVRS DSN('A.B.G^^^V^^') NOGDG ...

The extract programs use the latter method to most specifically match the retention and movement policies of CA-1. However, you can control what EDGCSVDS does by specifying one of the execution parameters--GDG, NOGDG, or BOTH.

EDGCSVDS optionally requires an execution parameter :

GDG Specify this parameter if the fully qualified data set names in the VPDD are only used to match the GDG base names. If any of the data set names match the standard data sets, use the BOTH parameter.

NONGDG

(Default) Specify this parameter if the fully qualified data set names in the VPDD never match a GDG base name. If any of the data set names match GDG base names, use the BOTH parameter.

BOTH Specify this parameter if the fully qualified data set names in the VPDD can potentially match either standard data set names or GDG base names. If you know more specifically what they match, consider either the GDG or NONGDG parameter.

Note: If the length of a data set name in the VPDD exceeds 35, only a NOGDG type VRS is created independently of the specified execution parameter.

Input Data Sets

EDGCSVDS has three input data sets:

Table 25. EDGCSVDS Input Data Sets

DDNAME	Description
VDS	VPDD unsorted (Required).
STORE	Storage location name input data set. (see Table 10 on page 33 for information on data set format). A maximum of 1000 vaults can be specified.
TMSPARM	The TMS PARMLIB options member (Optional).

Note: If you do not use the optional input data set, you must specify DD DUMMY for it.

Output Data Sets

EDGCSVDS creates an output data set containing the converted retention and storage location movement policies from the VPDD, the RDS, and the TMC volume records having the special meaning Julian dates and an error report file:

Table 26. EDGCSVDS Output Data Sets

DDNAME	Description
DEXTOUT	K-Records The LRECL of this file is overridden during program execution. The value in the JCL examples and samples is just an indication of the approximate record length.
ERROR	Information and error messages.
EXPRPT	Until Expired Retention Report. This file lists all occurrences of data sets in the VPDD that specify a retention of EXP.

Return Codes

EDGCSVDS sets the return code accordingly when it ends:

Table 27. EDGCSVDS Return Codes

Return code	Description
0	Normal processing.
4	Unable to support a VPDD policy option but K-Record is written.
12	Unable to create K-Record because the "V=" control statement is missing.
16	Unable to create K-Record because of a VPDD control statement error.

Messages

EDGCSVDS-00000 NO RMM EQUIVALENT FOR THIS TYPE OF POLICY, OPTION IGNORED AND CONTINUING - RC=04

Severity: Warning

Explanation: None.

System action: The job continues and sets a minimum return code of 4.

User response: None.

EDGCSVDS-00005 *ssss* STORE IS NOT DEFINED IN STORE DD FILE - RC=04

Problem determination: where

ssss Store location.

Severity: Warning

Explanation: None.

System action: The job continues and sets a minimum return code of 4.

User response: None.

EDGCSVDS-00010 V= CONTROL CARD MISSING FOR DATA SET: *dataset* - RC=12

Problem determination: where

dataset Data set name.

Severity: Warning

Explanation: None.

System action: The job continues and sets a minimum return code of 12.

User response: None.

EDGCSVDS-00015 A ZERO VAULT COUNT DETECTED - SET TO 1: *ccard* - RC=04

Problem determination: where

ccard Vault pattern description input data set record.

Severity: Warning

Explanation: None.

System action: The job continues and sets a minimum return code of 4.

User response: None.

EDGCSVDS-00020 VPDD CONTROL CARD ERROR: *ccard* - RC=16

Problem determination: where

ccard Vault pattern description input data set record.

Severity: Error

Explanation: None.

System action: The job continues and sets a minimum return code of 16.

User response: None.

EDGCSVDS-00025 ERROR dddddddddd IS AN UNSUPPORTED DATE FORMAT

Problem determination: where

ddddddddd

CA-1 parmlib options member TMS date format.

Severity: Error

Explanation: None.

System action: The job terminates with return code 16.

User response: Correct the option in the CA-1 parmlib member and rerun the job.

Policy Conversion Considerations

RMM is a policy driven system which offers a single point of control that is easy to manage. Policies are stored in RMM's CDS and multiple VRs for volume movement from one Storage Location to another and to SMS Tape Libraries are logically chained together.

Some of TMS/CA-1 policies are not compatible with RMM (see "Policy Conversion Differences" on page 65 for a detailed discussion).

Execution options are provided for the policy conversion programs to ease the retention and vaulting policy conversion process. In some instances, you may have to modify the policy conversion programs to meet your specific requirements. During the conversion process, consider exploiting RMM's policy management and simplicity of policy review, so as much as you can.

The policy conversion programs are written with the intention of helping you analyze your current TMS policies. This should help you determine what courses of action you need to take.

Early in the conversion process, you can bring up RMM in MANUAL mode after executing the policy conversion programs. Take this opportunity to review the converted policies via RMM's ISPF panels, or you can use the commands created in the EDGCNVT VRSCMDS output file.

If policy problems are found, make the appropriate changes to the RDS, the VPDD, and/or the policy conversion programs, and rerun the conversion process to recreate the RMM VRs.

Note: If the problems are few, they might be more quickly and easily resolved either via VRS updates through the RMM ISPF panels, or by editing the commands produced by EDGCNVT in the VRSCMDS output file.

RDS and VPDD Input Requirements

For correct processing by EDGCSRDS, and EDGCSVDS the data set statements in the RDS and VPDD must be correctly formatted. The sample job EDGJSRDS performs this reformatting for the RDS using a program called EDGCRFMT.

If data set names are found which are partial names, specified as an index, the program converts them to the DFSMSrmm equivalent. You must use EDGCRFMT in EDGJSRDS to reformat the RDS or the conversion programs will not work.

EDGCRFMT will also change the CA-1 special characters in a data set name to an equivalent DFSMSrmm character:

Table 28. Character conversion samples

CA-1	DFSMSrmm
-	*.**

Table 28. Character conversion samples (continued)

CA-1	DFSMSrmm
-.-	**.*
ABCDEFG-	ABCDEFG*.*
ABCDEFG.-	ABCDEFG.*
-ABCDEFG	*.*.*ABCDEFG
-.ABCDEFG	*.*.ABCDEFG
ABCDEFG-HIJKLMN	ABCDEFG*.*.*HIJKLMN
ABCDEFG.-HIJKLMN	ABCDEFG.*.*HIJKLMN
ABCDE@G.-	ABCDE%G.*
ABCDEG.-	ABCDE%G.*
ABCDE G.-	ABCDE%G.*
ABCDE?G.-	ABCDE*G.*

GDG vs. Non-GDG

TMS/CA-1 policy definitions do not differentiate between GDG and non-GDG data sets. It uses a single policy definition to identify both GDG and non-GDG data sets, i.e. DSN=A.B refers to data set "A.B" and GDG base "A.B".

RMM only offers the same capability when using its generic non-GDG format "A.B.*". RMM however, offers the ability to define a policy to cover the pseudo-GDG format and another type for only matching to GDG formats.

Note: The policy conversion programs by default only create the non-GDG format which RMM will apply to GDG and non-GDG data sets without differentiation as long as they match the data set name mask. Use the execution parameters of EDGCSRDS and EDGCSVDS to control the other options which are described in "Execution Parameters" on page 55 and "Execution Parameter" on page 61.

REDEFINED

TMS/CA-1 uses the REDEFINED option when you make changes to policies in order to recalculate data set retention needs. RMM always recalculates the retention needs so there is no need for the policy conversion programs to emulate the REDEFINED function.

BACKWARD

TMS/CA-1 uses the BACKWARD option to correct the volume position when you run into the condition where early cycles are prematurely removed from the Outcode Locations. RMM automatically recognizes when a volume is prematurely removed from the Storage Locations and will automatically schedule volumes be moved back to the appropriate Storage Location.

Policy Conversion Differences

This section discusses TMS/CA-1 policies that differ from RMM:

GDG CYCLE RETENTION

TMS/CA-1 cycle retention for GDGs does not consider the generation nor version number. DFSMSrmm uses the generation number in the same way that catalog processing uses the generation number.

DFSMSrmm can detect a wrap in the generation number, and, unlike catalog processing, can support multiple wraps for tape data sets that are no longer cataloged. As a result of this difference you may see slight differences in the decisions taken by TMS/CA-1 and DFSMSrmm.

DFSMSrmm also detects where a GDG generation has been recreated and will 'drop' the duplicate generation because it assumes you have recreated the data set intentionally. DFSMSrmm also supports a way to avoid the dropping of duplicate generations but this requires a usermod be installed (Refer to APAR OW56496 for details of the usermod).

MULTI-DATA SET MULTI-VOLUME RETENTION

DFSMSrmm can return a single volume in a multi-volume set of volumes to scratch - it does not need to operate always on the complete set of volumes.

Consequently, you may see DFSMSrmm return a volume to scratch that is still retained by TMS. Each data set in a multi-volume set can be retained individually using VRS.

If there are data sets in the volume set that have different retention requirements DFSMSrmm can attempt to release them earlier. For a volume to be released ALL data sets on that volume must no longer be retained by VRS.

If you use a controlling file to cause the complete set to be retained you may require additional VRS definitions under DFSMSrmm to cause the same volumes to be retained.

DFSMSrmm includes parmlib options to make RMM work in similar way as CA-1. Use the MOVEBY(SET) and RETAINBY(SET) options to manage multi-volume sets as an aggregate.

Alternatively you can benefit from DFSMSrmm releasing volumes early when the data is not required to be retained.

SEPDSN

TMS/CA-1 allows a policy definition using data set name prefix and allows another definition for a specific data set that has the same data set name prefix.

For example, you have "DSN=A.B-,SEPDSN" and "DSN=A.B.C" policies. The purpose is to handle data set "A.B.C" separately from any other data set with prefix "A.B". The policy conversion programs does not do anything special in this case for "DSN=A.B-,SEPDSN".

A generic data set name mask is created for data set having prefix "A.B" (VRS dsname is A.B*.*). In DFSMSrmm all data sets that match to a specific VRS are treated separately unless retention is by cycles. When cycles is used, all data sets with an identical name are managed as a group in order to retain the correct number of cycles.

Hints and Tips

This section describes some items that may be helpful during the conversion process:

- A media name in RMM is used to identify shape and not the media type. It must match an RMM VLPOOL definition in the EDGRMMxx member of PARMLIB.
- If you currently use TMS 4.9, and have been specifying expiration dates into the next century you need to do special activities to get the correct information to DFSMSrmm.

TMS sets the expiration date to 99366 to prevent expiration, and sets the correct 21st century date in the audit record it creates.

During conversion, EDGC4LDR issues a message for each occurrence of 99366 it encounters and you must obtain the expiration date from the audit records and use RMM CHANGEVOLUME volser EXPDT subcommand to set the correct date once you have started DFSMSrmm with the CDS created from conversion.

- TMS/CA-1 uses pre-defined Outcode Location names "LIBR" or "LIB" to delay volumes movement until the specified number of days is reached.

You can accomplish the same with RMM by defining a VRS with LOCATION(HOME) and use NEXTVRS to chain VRS definitions together. Do not use the DELAY option of VRS unless the delay is only to apply to the latest copy or cycle.

For example, you could have defined in the VPDD:

```
V=LIBR,CR=3
V=VLT1,CR=10
```

This indicates that the volumes are to be kept in the tape library for 3 days after creation before they are moved to Storage Location "VLT1".

The RMM equivalent to this, which is produced by the conversion programs is:

```
RMM ADDVRS DSN(...) DAYS COUNT(3) LOCATION(HOME) STORENUMBER(3) -  
  NEXTVRS(VRS00001)  
RMM ADDVRS NAME(VRS00001) DAYS COUNT(10) LOCATION(VLT1) -  
  STORENUMBER(10)
```

- Consider running the policy conversion programs to analyze your current tape management policies early in the planning process. This helps you understand existing policies and how they are converted for use with RMM.

Chapter 4. Conversion from TLMS

This chapter contains information about converting from TLMS to DFSMSrmm. For general conversion process questions refer to the Redbook "**Converting to DFSMSrmm from TLMS**" (SG24-6242).

If you do not have the Redbook "**Converting to DFSMSrmm from TLMS**" (SG24-6242), we recommend you to order a copy. You can check for the Redbooks on <http://www.redbooks.ibm.com> and search for 'RMM'.

The following steps summarize the recommended conversion process.

1. Review the Redbook "**Converting to DFSMSrmm from TLMS**" (SG24-6242), the rest of this chapter, and Chapter 9, "Building the DFSMSrmm CDS," on page 185.
2. Produce TLMS reports TLMS015 (Storage Location) and TLMS016 (Retention Management).
3. Evaluate existing retention management and volume movement policies.
4. Formulate EDGCDYNM and EDGCNVT control statements.
5. Define and customize the RMM system.
6. Start RMM subsystem in MANUAL mode to get familiarization with dialog and commands.
7. Execute the EDGCDYNM, EDGCNVT, and EDGCEBIN programs.
8. Load the RMM Control Data Set.
9. Use EDGUTIL to verify the RMM control data set, and to create the control record.
10. Start RMM subsystem in RECORD mode.
11. Manually correct any duplicate or GDG VRS definitions and save changes.
12. Execute RMM housekeeping process (VRSEL, EXPROC and DSTORE) to assign storage locations and bin numbers.
13. Compare RMM scratch and storage location reports with corresponding TLMS reports.
14. Correct any conversion problems and repeat steps 7 to 14.
15. Modify and Install any required installation exits, such as EDGUX100.
16. Start RMM in WARNING mode.
17. When no unexplained differences, start RMM in PROTECT mode and remove TLMS from the system.

The following table lists the conversion samples, programs and JCL for the extraction of data from TLMS.

Table 29. DFSMSrmm supplied conversion samples, programs, and JCL

Name	Location	Function
HLASM modules		
EDGCDYNM	WEB	Source for TLMS extract program
EDGCEBIN	LINKLIB	The EDGCEBIN program
EDGCNVT	LINKLIB	The EDGCNVT program
EDGCVOVL	LINKLIB	The EDGCVOVL program
EDGCNXIT	SAMPLIB	Source for sample exit for EDGCNVT
JCLs		
EDGJYPRE	WEB	JCL to compile and link EDGCDYNM
EDGJCNT	SAMPLIB	JCL to execute EDGCNVT and EDGCEBIN
EDGJDYNM	WEB	JCL to execute EDGCDYNM

Table 29. DFSMSrmm supplied conversion samples, programs, and JCL (continued)

Name	Location	Function
EDGJLOAD	SAMPLIB	JCL to execute EDGCVOVL, create and load CDS
EDGJVERR	SAMPLIB	JCL to create a CDS control record and execute EDGUTIL VERIFY processing

The sample JCL is set up to work in a "typical" installation. Some of the sample JCL is commented out so that you choose to use it or not as is your preference.

| **Note: Make sure to run EDGJYPRE prior to EDGJCNTV, in order to compile and link module**
| **EDGCDYNM.**

Functional Description

The data extraction programs are designed to extract relevant information from the TLMS VMF and RMF TLMS016 report (Retention Management File report). The information is used to create four sequential output files containing a series of records that are input to the EDGCNTV program before they are loaded into the DFSMSrmm CDS (see table below).

The LRECL of the output files is overridden during program execution. The value included in the JCL examples and samples is just an indication of the approximate record length.

Table 30. EDGCDYNM Output Data Sets and Records

DDNAME	Output description	Output record
DEXTLIBD	Data sets and volumes	<ul style="list-style-type: none"> • D (data set) record • L (volume) record
DEXTRBIN	Empty bin numbers in the storage locations and empty rack numbers in the tape library	<ul style="list-style-type: none"> • E (empty rack/bin) record
DEXTVRS	Retention and storage location management policies	<ul style="list-style-type: none"> • K (VRS) record
DEXTOWN	VRS and default owner information	<ul style="list-style-type: none"> • O (owner) record

Note that the data set and volume (D-type and L-type) records are sequence dependent - see Chapter 9, "Building the DFSMSrmm CDS," on page 185 for details.

Preparing to run the Extract Program

Before you begin the extraction process you should consider verifying the integrity of the TLMS VMF database to check for broken volume chains, expiration dates in error, or any other condition that can cause an error during data extraction.

You must run the appropriate TLMS utilities to verify the integrity of the database and correct the errors. The TLMS volume chaining verification system (TLMSVCVS) is a monitoring system that checks the validity of the links between volume base records. It explains the errors found in the VMF database and the actions taken for every error message.

Once you are sure that you have a valid VMF database, you can proceed to the next step.

The EDGCDYNM Program

This section describes the setup and control of the TLMS EDGCDYNM data extraction program.

EDGCDYNM supports all current releases of TLMS. However, the correct level of TLMS must be specified at the time you assemble the sample source code. You use the SYSPARM assembler option to specify:

TLMS53

for TLMS Release 5.3

TLMS54

for TLMS Release 5.4

TLMS55

for TLMS Release 5.5

See Figure 30 on page 77 for EDGCDYNM sample JCL.

Input Files

You need the following DD-statements for the input files:

Table 31. EDGCDYNM Input Files.

DDNAME	Description
SYSIN	Contains optional input parameters to control the execution of EDGCDYNM and the data extraction process.
TLMSVMF	Defines the TLMS VMF, which contains the tape volume and data set information to be converted.
TLMS016	Defines the TLMS Retention Selection Report file (TLMS report 016) which contains the volume retention and movement rules for converting to DFSMSrmm VRS definitions.

In addition you need TLMS report 015, which lists the storage locations you use with TLMS. You will use the information in the report to define the TLMSLOCN and DFRMMLOC statements in the SYSIN data set.

Output Files

You need the following DD-statements for the output files:

Table 32. EDGCDYNM Output Files.

DDNAME	Description
SYSPRINT	Contains informational and error messages associated with the data extraction process.
DEXTLIBD	Library and data set records for input to the EDGCNVT program.
DEXTOWN	Owner records for input to the EDGCNVT program.
DEXTRBIN	Empty rack and bin records for input to the EDGCNVT program
DEXTVRS	VRS records for input to the EDGCNVT program

Input Parameters

The EDGCDYNM program accepts a number of optional control statements input in the SYSIN file. These statements are identified by a keyword starting in position 1, followed by one or more parameters starting in position 9. The statements are fixed format.

Keywords and operands must start in the indicated positions. Optional keywords and operands can be left blank.

Statements can be supplied in any order. Comment statements are identified by an asterisk (*) in position 1.

If you do not specify the control statements in the exact format, the EDGCDYNM program attempts to use the data you enter, and results could be unpredictable, including ABEND0C7, and ignoring of bin counts.

Processing Options

The OPTIONS statement is used to control the way in which EDGCDYNM processes the TLMS VMF and RMF data.

The figure below shows the format of the statement,

OPTIONS	OUTSERV=00000000	VOLCHAIN=ccccccc	VOLLOCN=11111111
*	*	*	*
1	18	36	52

Figure 21. EDGCDYNM SYSIN: OPTIONS Control Statement

where:

- **00000000** specifies the processing for TLMS out-of-service volumes:
 - Ignore - specifies that out-of-service volumes are ignored and only empty rack records will be created
 - Process - specifies that out-of-service volumes are processed as standard scratch or nonscratch volumes
 - Loanloc - specifies a one- to eight-character loan location name. Out-of-service volumes are processed as scratch or nonscratch and have their loan location set to the specified value.
- **ccccccc** specifies the processing for volume chains:
 - Ignore - specifies that EDGCDYNM process the VMF sequentially and ignore any volume chaining information. All volumes will have a volume sequence number and count of 1.
This option ensures that all volumes are converted, but it loses all multi-volume count and chaining information.
 - Process - specifies that EDGCDYNM attempts to process volume chaining information to produce correct volume sequencing and counts.
Any errors in the VMF volume records may result in missing volumes in the converted data.
- **IIIIII** specifies the processing for the volume's home and current location:
 - Pass - specifies that EDGCDYNM does not convert the volume's current TLMS location. The DFSMSrmm current location and bin number are set to the TLMS location and cabinet or box number. Similarly, the TLMS RMF locations are forwarded into the corresponding DFSMSrmm VRS definitions.
The EDGCNVT program can be used to convert the TLMS information to DFSMSrmm data.
If TLMSLOCN statements are not supplied, the default processing option is PASS.
 - Procloc - specifies that EDGCDYNM ignore the volume's current TLMS location and set an DFSMSrmm home and current location of SHELF. The TLMS RMF locations are converted according to the TLMSLOCN specifications.
The DFSMSrmm housekeeping processes, VRSEL and DSTORE, are used after conversion to assign current locations and bin numbers according to the converted VRS definitions.
If TLMSLOCN statements are supplied, the default processing option is procloc.
 - Procbin - specifies that EDGCDYNM convert the TLMS locations according to the TLMSLOCN specifications. For those volumes in offsite locations, EDGCDYNM assigns a bin number. The bin numbers start at 1 for each location and are incremented by 1 for each volume assigned to that location.

Only one OPTIONS statement should be supplied; if not, the last statement encountered in the SYSIN file is used.

The default values for the OPTIONS statement are shown in the following example:

```
OPTIONS  OUTSERV=IGNORE  VOLCHAIN=PROCESS  VOLLOCN=PROCLOC
```

VRS Name Prefix

TLMS retention management policies can define multiple volume movements between storage locations. The equivalent DFSMSrmm function requires the use of VRS name definitions to chain together the separate movement definitions. The EDGCDYNAM extraction process automatically generates these VRS names according to an installation-defined prefix concatenated with an incremental number to make it unique.

The figure below shows the format of the statement,

VRSNAMEP	PREFIX=ppppp
*	*
1	17

Figure 22. EDGCDYNAM SYSIN: VRSNAMEP Control Statement

where:

- **ppppp** is a one-to-five-character alphanumeric field beginning with an alphabetic.
The generated VRS NAME is extended to eight characters by appending a unique numeric portion.
If a VRSNAMEP statement is not supplied, a default prefix of VRSN is used.

Only one VRSNAMEP statement should be specified; if not, the last statement encountered is used.

VRS Management Value Prefix

TLMS allows the specification of retention management policies through the use of special expiration dates. The equivalent RMM function requires the use of VRS management values and user exit EDGUX100. The EDGCDYNAM extraction process optionally can automatically generate these VRS management values according to an installation-defined prefix concatenated with the special Julian date. It also generates the corresponding VRS definition.

The figure below shows the format of the statement,

VRSMGMP	PREFIX=ppp
*	*
1	17

Figure 23. EDGCDYNAM SYSIN: VRSMGMP Control Statement

where:

- **ppp** is a one- to three-character alphanumeric field beginning with an alphabetic.
If a VRSMGMP statement is not supplied, special Julian dates are not processed and VRS management values or definitions are not produced.

Only one VRSMGMP statement should be specified; if not, the last statement encountered is used. For example, a control statement of:

```
VRSMGMP PREFIX=VDT
```

would produce, with a volume whose JCL expiry date was 98023, a VRS management value of:
VDT98023

and a VRS definition equivalent to the DFSMSrmm TSO subcommand:

```
ADDVRS DSN('VDT98023') LASTREFERENCEDAYS COUNT(23)
```

Note: The conversion process can only generate management values and VRS definitions for those special dates already in the VMF. For continued use of special dates the DFSMSrmm user exit EDGUX100 (or an appropriate ACS routine) must be used to set the management value for new tape data sets, and the associated VRS definitions must exist.

Storage Location Mapping

The TLMSLOCN control statement optionally specifies the mapping between TLMS location IDs and DFSMSrmm storage locations.

The figure below shows the format of the statement:

```

TLMSLOCN TLMSID=tt RMMLOC=IIIIIIII
*          *          *
1          17         27
  
```

Figure 24. EDGCDYNM SYSIN: TLMSLOCN Control Statement

where:

- **tt** is the TLMS storage location ID.
- **IIIIIIII** is the corresponding DFSMSrmm location name. Specify any one-to-eight-character name that is the location name you plan to use under DFSMSrmm. You can specify the TLMS location ID and optionally use EDGCNVT to translate it to the value you want to use.

One TLMSLOCN statement should be supplied for each TLMS location ID to be translated.

The following example illustrates a typical mapping:

```

TLMSLOCN TLMSID=TL RMMLOC=SHELF
TLMSLOCN TLMSID=DC RMMLOC=SHELF
TLMSLOCN TLMSID=CL RMMLOC=SHELF
TLMSLOCN TLMSID=D2 RMMLOC=D2
TLMSLOCN TLMSID=FS RMMLOC=FS
TLMSLOCN TLMSID=OS RMMLOC=OS
  
```

Storage Location Values

The DFRMMLOC control statement optionally defines an DFSMSrmm storage location. Its principal use is to define system-managed tape libraries (automated and manual) and the maximum number of bins to be used for DFSMSrmm storage locations (local, distant, and remote).

The figure below shows the format of the statement,

```

DFRMMLOC RMMLOC=IIIIIIII RMMTYP=rrrrrrrr BINS=nnnnnn
*          *          *          *
1          17         33         47
  
```

Figure 25. EDGCDYNM SYSIN: DFRMMLOC Control Statement

where:

- **IIIIIIII** is the DFSMSrmm location name. Specify any one- to eight-character name that is the location name you plan to use under DFSMSrmm. You can specify the TLMS location ID and optionally use EDGCNVT to convert it to the value you want to use.
- **rrrrrrrr** is the type of DFSMSrmm location, one of SHELF, STORE, MTL, or ATL.
- **nnnnnn** is the six-digit number, from 0 to 999999, of storage bin numbers (slots) in the DFSMSrmm storage location.

It should be specified only for shelf-managed storage locations. You do not need to specify a BINS value as EDGCDYNM generates its own numbers as required starting from 000001. Rack numbers are automatically generated for all volumes no matter where the volume is located.

If DFRMMLOC statements are not supplied, only the basic DFSMSrmm locations shelf, local, distant, and remote are recognized, and no empty bin numbers are generated for the DFSMSrmm storage locations.

The following example illustrates a typical mapping:

```
DFRMMLOC RMMLOC=FS      RMMTYP=STORE  BINS=001000
DFRMMLOC RMMLOC=D2      RMMTYP=STORE  BINS=000500
DFRMMLOC RMMLOC=OS      RMMTYP=STORE  BINS=
DFRMMLOC RMMLOC=ROBBY   RMMTYP=ATL
```

Volume Media Names

The VOLMEDIA control statement optionally specifies a volume prefix and the corresponding media name, type and home location. If VOLMEDIA statements are not supplied, a default media name of 3480 is assigned to all volumes.

Prefixes are scanned in the order of the control statements; to ensure correct processing, the most restrictive prefixes should be specified first.

The figure below shows the format of the statement,

VOLMEDIA VOLPFX=vvvvvv VMEDIA=mmmmmmmm VMTYPE=****t HOMELOC=11111111				
*	*	*	*	*
1	17	31	47	61

Figure 26. EDGCDYNM SYSIN: VOLMEDIA Control Statement

where:

- **vvvvvv** is the one- to six-character VOLSER prefix.
- **mmmmmmmm** is the corresponding DFSMSrmm media name. This should match the media names you specify on the VLPOOL and LOCDEF commands in the DFSMSrmm PARMLIB member.
- ******t** is the optional media type.

This is one of:

- REEL - non-cartridge-type tape reel
- 3480 - 3480 type cartridge (18 track)
- CST - 3490 Cartridge System Tape (36 track)
- ECCST - 3490 Enhanced Capacity Cartridge System Tape (36 track)
- HPCT - 3590 High Performance Cartridge Tape
- EHPCT - 3590 Extended high performance cartridge tape

and, if specified, overrides any values obtained from the TLMS VMF records.

- **11111111** is the optional home location name. If specified, this location name must match one of the DFSMSrmm location names, SHELF or any system-managed library name. Storage locations cannot be specified for a home location - EDGCNVT ignores them.

The default home location is SHELF.

Note: The VMEDIA names must match the DFSMSrmm VLPOOL definitions in the EDGRMMxx member of PARMLIB. Otherwise errors will occur when DFSMSrmm is active.

Skip Processing VOLSERS

The SKIPVOL control statement optionally provides a means to skip processing one or more VOLSERS in the TLMS VMF. This skipping might be required if, for example, there was an uncorrectable error in the TLMS data. The skipped volumes can then be manually added after conversion.

The figure below shows the format of the statement,

```
SKIPVOL  VOLSER=vvvvvv
*          *
1          17

      OR:

SKIPVOL  VOLRNG=vvvvvv-vvvvvv
*          *      *
1          17      24
```

Figure 27. EDGCDYNM SYSIN: SKIPVOL Control Statement

where:

- **VOLSER** specifies a single VOLSER.
- **VOLRNG** specifies a range of VOLSERS.
- **vvvvvv** is a six-character VOLSER.

Default Volume Owner ID

DFSMSrmm requires that all nonscratch volumes be assigned an owner, represented by a one- to eight-character owner identifier. Because this function is not present in TLMS, a default owner ID must be provided. The DEFOWNER statement lets you provide an owner ID different from the default of LIBRARIAN.

The figure below shows the format of the statement,

```
DEFOWNER 00000000
*          *
1          10
```

Figure 28. EDGCDYNM SYSIN: DEFOWNER Control Statement

where:

- **00000000** is a one- to eight-character alphanumeric field beginning with an alphabetic.

If a DEFOWNER statement is not supplied, a default owner of LIBRARIAN is used.

Only one DEFOWNER statement should be specified; if not, the last statement encountered is used.

VRS Owner ID

DFSMSrmm requires that all VRS definitions be assigned an owner, represented by a one- to eight-character owner identifier. TLMS provides a 25-character authorization field in the RMF report. EDGCDYNM can attempt to use this field by extracting the first eight non-blank characters to form an owner ID. This ID may not always be suitable or valid, so the VRSOWNER statement gives the option of providing a fixed default owner ID.

The figure below shows the format of the statement,

```

VRSOWNER 00000000
*        *
1        10

```

Figure 29. EDGCDYNM SYSIN: VRSOWNER Control Statement

where:

- **00000000** is a one-to-eight-character alphanumeric field beginning with an alphabetic. A value of AUTH indicates that the TLMS authorization field should be used.

If a VRSOWNER statement is not supplied, the TLMS authorization field is used if valid; otherwise the default volume owner ID is used.

Only one VRSOWNER statement should be specified; if not, the last statement encountered is used.

Program Execution

The figure below shows the sample JCL to execute the extract process for TLMS. See sample EDGJDYNM for the supplied sample JCL.

```

//JOBNAME JOB .....
//*
//DEXTRACT EXEC PGM=EDGCDYNM
//*STEPLIB DD DISP=SHR,DSN=rmm.load
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//*
//*      TLMSVMF is the TLMS volume master file
//TLMSVMF DD DISP=SHR,DSN=tlms.vmf
//*
//*      TLMS016 is the retention selection report (Report 016)
//TLMS016 DD DISP=SHR,DSN=tlms.tlms016.report
//DEXTLIBD DD DISP=(,CATLG),DSN=STSGWD.EDGCDYNM.DEXTLIB.DATA,
//          RECFM=VB,LRECL=700,AVGREC=K,SPACE=(700,(50,20),RLSE)
//DEXTOWN DD DISP=(,CATLG),DSN=STSGWD.EDGCDYNM.DEXTOWN.DATA,
//          RECFM=VB,LRECL=512,AVGREC=K,SPACE=(512,(1,1),RLSE)
//DEXTRBIN DD DISP=(,CATLG),DSN=STSGWD.EDGCDYNM.DEXTRBIN.DATA,
//          RECFM=VB,LRECL=512,AVGREC=K,SPACE=(512,(10,5),RLSE)
//DEXTVRS DD DISP=(,CATLG),DSN=STSGWD.EDGCDYNM.DEXTVRS.DATA,
//          RECFM=VB,LRECL=512,AVGREC=K,SPACE=(512,(1,1),RLSE)
//SYSIN DD *
OPTIONS OUTSERV=IGNORE VOLCHAIN=PROCESS VOLLOCN=PROCLOC
VRSNAMEP PREFIX=VRSN
VRSMGMTP PREFIX=VDT
TLMSLOCN TLMSID=TL RMMLOC=LIBRARY
TLMSLOCN TLMSID=DC RMMLOC=LIBRARY
TLMSLOCN TLMSID=CL RMMLOC=LIBRARY
TLMSLOCN TLMSID=D2 RMMLOC=LOCAL
TLMSLOCN TLMSID=FS RMMLOC=REMOTE
TLMSLOCN TLMSID=OS RMMLOC=DISTANT
DFRMMLOC RMMLOC=LOCAL RMMTYP=SHELF BINS=001000
DFRMMLOC RMMLOC=DISTANT RMMTYP=SHELF BINS=002000
DFRMMLOC RMMLOC=REMOTE RMMTYP=SHELF BINS=001000
/*

```

Figure 30. EDGCDYNM Sample JCL

Program Limitations

The following are the current program implementation restrictions for EDGCDYNM:

- A maximum of 100 TLMSLOCN statements are allowed.

- A maximum of 100 DFRMMLOC statements are allowed.
- A maximum of 100 SKIPVOL statements are allowed.
- A maximum of 100 VOLMEDIA statements are allowed.

Return Codes

A return code is set as follows:

- | | |
|----|---|
| 0 | Normal completion. No errors or warnings detected. |
| 4 | One or more warning messages issued. A valid, converted CDS is produced, but the warning messages should be reviewed to ensure that results are as expected. For example, default values may have been used for conversion when not intended. |
| 8 | One or more error messages issued. The CDS may contain invalid data. The errors should be corrected and the conversion process rerun. |
| 16 | An unrecoverable error has been encountered causing the conversion process to end early. The errors must be corrected and the conversion process rerun. |

Messages

EDGCDYNAM-00002 END OF PROGRAM EDGCDYNAM. HIGHEST RETURN CODE WAS *xx*

Problem determination: where

xx Highest return code set by the program.

Severity: Information

Explanation: The EDGCDYNAM program has completed execution.

System action: None

User response: If a return code greater than 0 is set, the previous messages should be reviewed and any errors corrected before rerunning the program.

EDGCDYNAM-00010 TLMS DATA EXTRACTION PARAMETERS USED:

Severity: Information

Explanation: The SYSIN control statements are listed following this message for information purposes.

System action: None

User response: None

EDGCDYNAM-00011 *ERROR* IN PREVIOUS CONTROL STATEMENT - PARAMETERS IGNORED.

Severity: Error

Explanation: An error was encountered in the previous control statement. This could include an unidentified keyword or keyword value, or a value out of range.

System action: The indicated control statement is ignored, and default values are assumed where appropriate.

User response: Correct the control statement and rerun the data extraction program.

EDGCDYNAM-00012 *ERROR* UNIDENTIFIED DFRMM STORAGE LOCATION - *xxxxxxxx*

Problem determination: where

xxxxxxxx
 DFSMSrmm storage location defined in a TLMSLOCN statement

Severity: Error

Explanation: The identified storage location name was specified on a TLMSLOCN statement, but there is no corresponding DFRMMLOC statement.

System action: The TLMS. location IDs corresponding to the identified DFSMSrmm location will be defaulted to a location of LIBRARY.

User response: Correct the corresponding TLMSLOCN statements and/or supply appropriate DFRMMLOC statements and rerun the data extraction process.

EDGCDYNAM-00013 **CRITICAL ERROR UNIDENTIFIED RECORD TYPE=*t* IN THE VMF FILE**

Problem determination: where

t record type identifier

Severity: Terminal

Explanation: An unidentified record type was encountered in the VMF data set.

System action: The process is ended with a return code of 16.

User response: Correct the TLMSVMF data set records and rerun the data extraction program. EDGCDYNAM is designed to run against the actual or image copy of the VMF. Do not use a backup copy taken using the TLMS utility as input.

EDGCDYNAM-00014 **CRITICAL ERROR VOLUME *xxxxxx* IS NOT IN ANY DEFINED VOLUME RANGE**

Problem determination: where

xxxxxx volume serial number

Severity: Terminal

Explanation: Specified volume is not in any defined volume range.

System action: The process is ended with a return code of 16.

User response: Correct the TLMSVMF control records and rerun the data extraction program. EDGCDYNAM is designed to run against the actual or image copy of the VMF. Do not use a backup copy taken using the TLMS utility as input.

EDGCDYNAM-00020 *WARN* DUPLICATE RMF DSNAME IGNORED - *t ddd.....ddd jjjjjjj*

Problem determination: where

t type of RMF entry, Full, Partial or Qualified

ddd.....ddd

44-character duplicate data set name

jjjjjjj optional 8-character job name qualifier

Severity: Warning

Explanation: An RMF entry with an identical data set and job name combination to a previous entry was encountered.

System action: The indicated RMF entry is ignored.

User response: Review the generated VRS definitions to ensure these produce the same results as the RMF specifications.

EDGCDYNAM-00021 *ERROR* UNSUPPORTED RETENTION COMBINATION IGNORED - *ddd.....ddd jjjjjjj*

Problem determination: where

ddd.....ddd

44-character duplicate data set name

jjjjjjj optional 8-character job name qualifier

Severity: Error

Explanation: An RMF entry with an unsupported combination of retention types was encountered. RMM chained VRS definitions must have the same retention type, for example, DAYS or CYCLES.

System action: The indicated RMF entry is ignored.

User response: Review the indicated RMF entry to determine the intended management policy. Manually convert this policy to the nearest equivalent RMM VRS definition.

EDGCDYNAM-00022 *ERROR* UNIDENTIFIED TLMS LOCATION=*ll* RMF=*rrr.....rrr*.

Problem determination: where

ll unidentified TLMS location

rrr.....rrr

data set name / job name identifying the RMF definition

Severity: Error

Explanation: An RMF entry with an unidentified location identifier was encountered. If mapping of TLMS locations to DFSMSrmm locations is required, the locations must be defined in TLMSLOCN statements.

System action: The unidentified location ID is translated to an DFSMSrmm location of HOME.

User response: Review the indicated RMF entry and supply an appropriate TLMSLOCN statement to map the TLMS ID to an DFSMSrmm location. Alternatively, set the location processing option (VOLLOCN on the OPTIONS statement) to PASS and use the EDGCNVT program to map the location identifiers.

EDGCDYNAM-00023 *WARN* ELAPSED DAYS RETENTION IGNORED - *ddd.....ddd* *jjjjjjj*

Problem determination: where

ddd.....ddd

44-character duplicate data set name

jjjjjjj

optional 8-character job name qualifier

Severity: Warning

Explanation: An RMF entry with an unsupported combination of retention types was encountered (CYCLES with ELAPSED DAYS/WHILECATALOG). EDGCDYNAM has attempted to produce a valid K-Record for input to EDGCNVT.

System action: The indicated RMF entry has been processed with the following conditions:

- If a retention of DAYS is followed by CYCLES, the DAYS specification is converted to an RMM DELAY specification.
- If a retention of CYCLES is followed by ELAPSED DAYS/WHILECATALOG, the DAYS specification is ignored and a retention of WHILECATALOG used.

User response: Review the indicated RMF entry to determine the intended management policy. If necessary, manually update the VRS produced by the conversion process.

EDGCDYNAM-00024 *WARN* CRASH PROTECTED DATA SET *ddd.....ddd* DETECTED ON NON-SCRATCH VOLUME *vvvvvv*. PLEASE CHECK DATA SET NAME.

Problem determination: where

ddd.....ddd

truncated 21 character data set name

vvvvvv

6 character volume serial number

Severity: Warning

Explanation: A TLMS crash protected data set was detected on a non scratch volume. It is assumed that crash protection applies only to the first data set on a multi-file and/or multi-volume set.

System action: The TLMS prefix 'TLMSII-CRASH-PROTECTED-' has been removed from the start of the data set name. The data set name may have been truncated by TLMS. If EDGCDYNM recognizes the data set name is not truncated, the volume is marked as recorded by O/C/EOV, else the text 'PLEASE CHECK DATA SET NAME' appears in the message and volume is not marked as recorded by O/C/EOV.

User response: If the data set name is not truncated you can continue with the conversion. If it may be truncated you must check for the correct data set name and if necessary correct the data set name in TLMS and rerun EDGCDYNM to correct the data set name, or use the DFSMSrmm CHANGEVOLUME command to correct the data set name once DFSMSrmm is started with the converted CDS.

EDGCDYNM-00026 *WARN* PARMLIB OPTION RETAINBY(SET) REQUIRED.

Severity: Warning

Explanation: One of TLMS retention types has been discovered.

System action: None.

User response: Specify parameter RETAINBY(SET) in the EDGRMMxx parmbib member.

EDGCDYNM-00030 *WARN* INSUFFICIENT BIN NUMBERS SUPPLIED FOR LOCATION - xxxxxxxx.
USED=uuuuuu SUPPLIED=sssss

Problem determination: where

xxxxxxx

storage location

uuuuuu number of bin numbers allocated to volumes

sssss count of bin numbers supplied

Severity: Warning

Explanation: The PROCBIN processing option was specified on the OPTIONS statement to force allocation of bin numbers to volumes in storage locations. Either no or insufficient bin numbers were specified on the DFRMMLOC statement defining the location.

System action: Bin numbers will be correctly assigned, but no empty bins will be generated.

User response: Either supply DFRMMLOC statements with appropriate bin numbers or manually add empty bins after conversion using the DFSMSrmm dialog or ADDBIN TSO subcommand. If no empty bins are generated or added, the RMM housekeeping process DSTORE may subsequently fail through lack of empty bins.

EDGCDYNM-00031 *WARN* NO TLMSLOCN STATEMENTS SUPPLIED - LOCATION PROCESSING REQUESTED.

Problem determination:

Severity: Warning

Explanation: You requested, or used by default, the PROCLOC option but did not specify any TLMSLOCN statements.

System action: A processing option of PASS is forced; no mapping of TLMS storage locations will be performed.

User response: Either supply TLMSLOCN statements with appropriate mappings or specifically request a processing option of PASS on the OPTIONS statement.

EDGCDYNM-00032 *ERROR* UNIDENTIFIED TLMS LOCATION=ll VOLSER=vvvvvv

Problem determination: where

ll unidentified TLMS location

vvvvvv VOLSER

Severity: Error

Explanation: A volume entry with an unidentified TLMS location identifier was encountered. If mapping of TLMS locations to DFSMSrmm locations is required, the locations must be defined in TLMSLOCN statements.

System action: The unidentified location ID is translated to an DFSMSrmm location of SHELF.

User response: Review the indicated volume entry and supply an appropriate TLMSLOCN statement to map the TLMS ID to a DFSMSrmm location. Alternatively, set the location processing option (VOLLOCN on the OPTIONS statement) to PASS and use the EDGCNVT program to map the location identifiers.

EDGCDYNM-00040 *WARN* ZERO VOLUME SEQUENCE NUMBER SET TO ONE - VOLSER=vvvvvv

Problem determination: where

vvvvvv VOLSER

Severity: Warning

Explanation: A nonscratch volume entry with a zero volume sequence number was encountered. This could cause errors in EDGCNVT and/or DFSMSrmm processing.

System action: The zero volume sequence number is set to 1 to prevent possible later errors.

User response: Review the indicated volume entry to ensure that it contains valid information.

EDGCDYNM-00092 *ERROR* INSUFFICIENT NUMBER OF ENTRIES IN VRSMTAB.

Problem determination:

Severity: Error

Explanation: The message is issued when more than the specified VRS management values are in use. VRS management values are built during conversion from the special keyword dates detected in the VMF. A VRS management value is built for each unique date detected.

System action: A VRS management value K record is built but duplicates are not tracked.

User response: Correct the data extraction program EDGCDYNM: find the line marked as "=====> VRSMTAB SIZE DEFINITION" and increase the number of entries. Then assemble it and rerun the program. If the number of entries will not be increased, the load of the DFSMSrmm CDS will fail because of duplicate K-Records.

EDGCDYNM-00093 *ERROR* INCORRECT CHAIN POINTER(S) ENCOUNTERED. RECORDS SKIPPED

=nnnnnnnn

Problem determination: where

nnnnnnnn

number of records skipped due to errors in chain pointers

Severity: Error

Explanation: A VMF entry with an incorrect TLMS chain pointer has been encountered.

System action: The volume records pointed to by the incorrect chain pointers are skipped.

User response: Run the TLMSV CVS utility to correct chain pointers and rerun the data extraction program.

EDGCDYNM-00094 RUN TLMSV CVS UTILITY TO CORRECT CHAIN POINTERS

Problem determination:

Severity: Information

Explanation: This message will follow the EDGCDYNM-00093 message.

System action: None

User response: Run the TLMSV CVS utility to correct chain pointers and rerun the data extraction program.

EDGCDYNM-00095 EDGCDYNM INPUT RECORD COUNTS:

Problem determination:

Severity: Information

Explanation: This message will be followed by one or more EDGCDYNM-00096 messages indicating the number of records read of a particular record type.

System action: None

User response: None

EDGCDYNM-00096 NUMBER OF *rrrrrrrrrrrr* RECORDS READ = *nnnnnnnnnn*

Problem determination: where

rrrrrrrrrrrr
type of record

nnnnnnnnnn
number of records read

Severity: Information

Explanation: Information

System action: None

User response: None

EDGCDYNM-00097 EDGCDYNM OUTPUT RECORD COUNTS:

Problem determination:

Severity: Information

Explanation: This message will be followed by one or more EDGCDYNM-00098 messages indicating the number of records written for a particular record type.

System action: None

User response: None

EDGCDYNM-00098 NUMBER OF *rrrrrrrrrrrr* RECORDS WRITTEN = *nnnnnnnnnn*

Problem determination: where

rrrrrrrrrrrr
type of record

nnnnnnnnnn
number of records written

Severity: Information

Explanation: Information

System action: None

User response: None

TLMS Conversion Considerations

This section describes the usage considerations for the TLMS to RMM conversion tools.

Storage Locations

RMM supports an almost unlimited number of storage and loan locations. You should be able to convert your TLMS storage locations to DFSMSrmm without hitting a limit caused by their number. In

DFSMSrmm there are three built-in storage locations which are all managed by bin number (slot), but by using LOCDEF commands in the DFSMSrmm parmlib you can use installation defined names, including those you are familiar with under TLMS, and decide whether to have them shelf managed or not.

When you convert to DFSMSrmm you have the choice to stay using the storage location names you currently use with TLMS, or of using different names. You may wish to use different names because DFSMSrmm allows up to eight character names rather than just two characters. If you decide to use the DFSMSrmm built-in storage location, LOCAL, DISTANT, and REMOTE you do have to translate location names and restricted to three storage locations. You can use a combination of built-in and installation defined storage locations requiring some translation of names. When you decide to use different names under DFSMSrmm, you can translate the TLMS storage location names to DFSMSrmm storage location names either during the EDGCDYNM run or during the EDGCNVT run. To translate the names with EDGCDYNM use the TLMSLOCN RMMLOC keyword to specify a different name. To translate the names with EDGCNVT use the LOCDEF SYSIN statement and the IF STORLOC statement to specify a different name.

RMM also supports an almost unlimited number of loan locations. If you wish to use these as a replacement for any of the TLMS storage locations, you can do so by using the LOCDEF statement in EDGCNVT to identify the location as a type of LOAN. When running EDGCDYNM you should treat it as a storage location and use TLMSLOCN and DFRMMLOC statements.

Shelf Management

All volumes defined to DFSMSrmm have a rack number assigned to provide shelf management while in the data center, and to provide a unique external identifier for the volume. EDGCDYNM and EDGCNVT use the TLMS volume serial number as the DFSMSrmm rack number.

Shelf management in DFSMSrmm storage locations is optional, but if provided it is done using bin numbers. Each bin number represents a single slot to store a volume. The EDGCDYNM OPTIONS statement provides you control over what the conversion process does for bin numbers. You know, from TLMS015 report, whether your storage locations are managed as cabinets, boxes or are undefined storage. You have to decide how best to convert this into DFSMSrmm. The OPTIONS VOLLOCN=PASS and VOLLOCN=PROCBIN options both pass on bin information; PASS passes on the cabinet and box numbers, and PROCBIN generates bin numbers starting from 000001. Here are some options:

- Specify VOLLOCN=PROCLOC and use DFSMSrmm inventory management to assign bin numbers and locations.
- Use VOLLOCN=PROCBIN and have bin numbers generated for you. These are passed from EDGCDYNM to EDGCNVT, and, although they are simply numeric bin numbers starting from 000001 through 999999, you can use them either with the built-in or installation defined storage locations. If you decide on non-shelf managed storage locations the numbers are simply ignored by EDGCNVT.
- Specify VOLLOCN=PASS and use the existing cabinet and box numbers under DFSMSrmm. When you use this option you may have to make some changes to the sample extract and conversion programs, as the cabinet and box numbers are only 4 digits and DFSMSrmm requires 6 digits.

Cabinet numbers could be expanded to six digits and used as bin numbers with installation defined storage locations. For example:

```
CCnn could be expanded to either  
CC00nn or  
CCnn00
```

and you could define bin numbers as CC0001 through CC0099 or CC0100 to CC9900 always leaving the last two digits as zero (00). A small change to EDGCDYNM just before label PVMFS120 could insert '00' into the cabinet number. The EDGCDYNM program builds a 6 character bin number by combining the 2 digit location ID with the 4 digit slot number. You can customize this code if it is not what you want. DFSMSrmm shelf management is very similar to the cabinet management provided by TLMS.

For box numbers, there is no box slot number kept in the VMF, but you could process it in a similar way to the cabinet numbers and generate the last two digits in numerical sequence for each volume found in the same box. For example:BBBB could be expanded to BBBBnn and you define bin numbers BBBB01 through BBBB99 to DFSMSrmm and be sure to define the location as an installation defined storage location using LOCDEF. Remember that DFSMSrmm does not provide box management and can only use the box/slot numbers as conventional open shelving.

Retention Criteria

Sample TLMS RMF to DFSMSrmm Conversion

This section illustrates a sample conversion of various RMF retention combinations. Data in the figure below is converted to the RMM commands shown in Figure 33 on page 86. Actually, RMM CDS records are produced rather than the TSO subcommands, but you can use the EDGCNVT program VRSCMDS DD to generate the ADDVRS commands so that you can easily check what is generated for DFSMSrmm.

This example assumes that the following TLMSLOCN statements are used to map the TLMS locations to DFSMSrmm equivalents:

```

TLMSLOCN TLMSID=DC RMMLOC=SHELF
TLMSLOCN TLMSID=TL RMMLOC=SHELF
TLMSLOCN TLMSID=CL RMMLOC=SHELF
TLMSLOCN TLMSID=D2 RMMLOC=D2
TLMSLOCN TLMSID=FS RMMLOC=FS
TLMSLOCN TLMSID=OS RMMLOC=OS

```

ANY COMPANY		TAPE RETENTION						
CA-DYNAM/TLMS 5.3 9012TL530								
Q-IND	I-----	DATA SET NAME OR QUALIFIER				-----I		
		DATA CENTER LOCATION				1ST OFF-SITE		
		T	ID	NUM	VER	T	ID	NUM
		-	--	----	----	-	--	----
DEFAULT								
		1	DC					
F		ACSS.DUMP.LOADDATA.D0001						
		2	DC	0090				
Q		ACSS.DUMP.RAWDATA.						
		2	DC	0090				
P		ADB2.ARCHLOG				5	FS	0001
		6	DC					
Q		ADB2.IMAGCOPY.				5	FS	0001
		6	DC					
Q		PCSS.DFDSS.				4	FS	0002
		6	DC					
Q		STSG.VTAM.BACKUP.				4	FS	0002
		6	DC					

Figure 31. Sample Retention Selection Report for Conversion (left side)

SELECTION RECORDS

TLMS016 PAGE 1
13 OCT 1992 11.50.10

JOB-NAME		ADD DATE		I---- AUTHORIZATION ----I				
2ND OFF-SITE		3RD OFF-SITE		4TH OFF-SITE		5TH OFF-SITE		
T	ID	NUM	T	ID	NUM	T	ID	NUM
5	TL	0013						
5	TL	0020						
1	TL							
4	D2	0005	1	TL				

Figure 32. Sample Retention Selection Report for Conversion (right side)

```

ADDVRS DSN('**') WHILECATALOG LOC(HOME) COUNT(99999) -
OWNER(STSG002)

ADDVRS DSN('ACSS.DUMP.LOADDATA.D0001') DAYS LOC(HOME) -
COUNT(90) OWNER(SDAC005)

ADDVRS DSN('ACSS.DUMP.RAWDATA.**') DAYS LOC(HOME) -
COUNT(90) OWNER(SDAC005)

ADDVRS DSN('ADB2.ARCHLOG*.**') DAYS LOC(FS) -
COUNT(1) OWNER(SDAC005) STORENUMBER(1) NEXTVRS(N1)
ADDVRS NAME(N1) DAYS COUNT(13) LOCATION(HOME) -
STORENUMBER(13) OWNER(SDAC005)

ADDVRS DSN('ADB2.IMAGCOPY.**') DAYS LOCATION(FS) -
COUNT(1) OWNER(SDAC005) STORENUMBER(1) NEXTVRS(N2)
ADDVRS NAME(N2) DAYS COUNT(20) LOCATION(HOME) -
OWNER(SDAC005) STORENUMBER(20)

ADDVRS DSN('PCSS.DFDSS') GDG CYCLES LOCATION(FS) -
COUNT(2) OWNER(SDAC018) STORENUMBER(2) NEXTVRS(N3)
ADDVRS NAME(N3) WHILECATALOG LOCATION(HOME) -
OWNER(SDAC018)

ADDVRS DSN('STSG.VTAM.BACKUP.**') CYCLES -
LOCATION(FS) COUNT(2) OWNER(STSG001)
STORENUMBER(2) NEXTVRS(VRSSTSG1)
ADDVRS NAME(VRSSTSG1) OWNER(STSG001) COUNT(5) CYCLES -
LOCATION(D2) STORENUMBER(5) NEXTVRS(VRSSTSG2)
ADDVRS NAME(VRSSTSG2) OWNER(STSG001) WHILECATALOG -
LOCATION(HOME)

```

Figure 33. Sample Converted VRS Definitions

Duplicate RMF Definitions

To avoid VRS definitions with duplicate data set and job names being generated, EDGCDYNM only processes the **FIRST** occurrence of a data set and job name combination and ignores all others.

A warning message is produced for each ignored data set. These should be reviewed after conversion to ensure the desired retention criteria are being applied.

GDG and Non-GDG Processing

TLMS allows the specification of full or qualified data set names, whereas RMM allows full generic specification qualified by GDG or non-GDG.

The conversion process produces VRS definitions for non-GDG data sets. These should be reviewed after conversion for any data sets that may be GDGs. This is particularly important if CYCLES control is being used. A NOGDG VRS definition can retain GDG data sets but only if the data set mask is generic and includes a generic character that covers the last qualifier of the GDG data set; For example:

```
Q  A.B.
```

```
is translated to  
DSN('A.B.**')
```

which matches to both 'A.B.C' a non-GDG, and 'A.B.G0010V00' a GDG.

Special Expiration Dates

While the EDGCDYNM program allows for processing of **existing** special expiration dates and the creation of VRS management values and definitions, it is the installation's responsibility to ensure that:

1. any new retention criteria are defined as VRS management values and VRS definitions. **AND**
2. either an appropriate DFSMS ACS routine or RMM installation exit EDGUX100 are installed and active to set VRS management values from special expiration dates.

Multi-Data Set Multi-Volume Retention

DFSMSrmm can return a single volume in a multi-volume set of volumes to scratch - it does not need to operate always on the complete set of volumes. Consequently, you may see DFSMSrmm return a volume to scratch that is still retained by TLMS.

Each data set in a multi-volume set can be retained individually using VRS. If there are data sets in the volume set that have different retention requirements DFSMSrmm can attempt to release them earlier. For a volume to be released ALL data sets on that volume must no longer be retained by VRS.

If you use a controlling file to cause the complete set to be retained you may require additional VRS definitions under DFSMSrmm to cause the same volumes to be retained. DFSMSrmm includes parmlib options to make RMM work in similar way as TLMS. Use the MOVEBY(SET) and RETAINBY(SET) options to manage multi-volume sets as an aggregate. Alternatively you can benefit from DFSMSrmm releasing volumes early when the data is not required to be retained.

Scratch Volumes

DFHSM Tape Processing

TLMS and RMM process the release of DFHSM tapes differently. When TLMS is informed via the ARCTVEXT exit that a DFHSM volume is no longer required, it changes the volumes creating jobname to a different pre-defined name (by default, one of HSMCOPY, HSMSCRDC or HSMSCROF). Jobname qualified RMF definitions are then used to typically retain the data set for 1 day before being released to the scratch pool.

In contrast, when RMM is informed by DFHSM, the volume is immediately marked 'pending release' and will be returned to the scratch pool following the next run of the housekeeping EXPROC process.

The result is that RMM will release DFHSM tapes at least one day sooner than TLMS. This must be taken into account when comparing scratch reports for the two systems.

Exported Logical Volumes & Stacked Volumes

EDGCDYNM detects exported logical volumes in the VMF and converts those as logical volumes. It also detects stacked volumes which are being used or have been used by export processing. However it cannot detect all stacked volumes. Therefore, it is best to use the EDGCNVT SYSIN file IF VOLRANGE with TYPE STACKED to define the complete range of stacked volumes.

To avoid problems with bin numbers for exported stacked volumes not matching the cabinet/box slot numbers you should avoid use of bin numbers in DFSMSrmm. Use the EDGCNVT SYSIN LOCDEF statement with NOBINS. If you instead use BINS, when you run EDGUTIL with MEND(VOLUME) to enable stacked volume support, the bin number from the first logical volume in sequence will be used instead of the existing assigned value from the stacked volume.

Parallel Running

UNCATALOG Option

During parallel running, it is strongly recommended that the RMM initialization parameter UNCATALOG in the EDGRMMxx PARMLIB member be set to N. This is especially so if the TLMS retention default is WHILECATALOG. Otherwise, differences in VRS and RMF definitions could result in RMM releasing and uncatalog volumes which would normally have been kept by TLMS.

Note that with PTF UY98946 applied to RMM, the default is 'N' when running in 'Record' or 'Warning' modes.

Sequence Dependency

When running DFSMSrmm with TLMS 5.5 and later releases, be sure to start DFSMSrmm before TLMS or to include the EDGSSSI program name in IEFSSNxx. Both products update LPA to get control during OPEN. The updates must be done in the correct order for processing to work correctly.

Chapter 5. Conversion from Manual Management to DFSMSrmm

This chapter contains information about converting from Manual Management using the ICF catalog to DFSMSrmm. For general conversion process questions refer to the Redbook "Converting to DFSMSrmm from ICF-usercatalog" (SG24-6244) .

If you do not have the Redbook "Converting to DFSMSrmm from ICF-usercatalog" (SG24-6244) , we recommend you to order a copy. You can check for the Redbooks on: <http://www.redbooks.ibm.com> and search for "RMM".

Following table lists the tools provided to aid conversion from Manual Management.

Table 33. DFSMSrmm-supplied conversion samples, programs, and JCL

Name	Location	Function
EDGCVTC0	WEB	Create LISTCAT commands by using the LISTCAT UCAT command
EDGCVTC4	WEB	Create ADDRACK, ADDVOLUME and CHANGEVOLUME commands to build an RMM CDS from the ICF-CATALOG input
EDGCVTC5	WEB	Create additional ADDVOLUME and CHANGEVOLUME commands
EDGCVTC6	WEB	Create CHANGEVOLUME commands to build the volume chains
EDGJVTC1	WEB	JCL to convert from manual management part 1
EDGJVTC2	WEB	JCL to convert from manual management part 2
EDGJVTC3	WEB	JCL to convert from manual management part 3
EDGCNVT	LINKLIB	The EDGCNVT program
EDGCNXIT	SAMPLIB	Source for sample exit for EDGCNVT
EDGCEBIN	LINKLIB	The EDGCEBIN program
EDGJCNT	SAMPLIB	JCL to execute EDGCNVT and EDGCEBIN
EDGCVOVL	LINKLIB	The EDGCVOVL program
EDGJLOAD	SAMPLIB	JCL to execute EDGCVOVL, create and load CDS
EDGJVERR	SAMPLIB	JCL to create a CDS control record and execute EDGUTIL VERIFY processing

Sample JCL

EDGJVTC1

- Create a sequential output file by using LISTCAT UCAT to list all user catalog connector entries. Using the EDGCVTC0 REXX procedure.
- Create a sequential output file by using LISTCAT from all of your available catalogs.
- Pre-allocate a new EXTRACT and MESSAGE data set.

EDGJVTC2

- Perform inventory management - EDGHSKP.
- Extract the remaining catalog entries with the RMM extract file by using the EDGCVTC4 REXX procedure.

- Create ADDRACK and ADDVOLUME commands, if the volume is not stored in the RMM control data set, using the EDGCVTC5 REXX procedure.
- Create CHANGEVOLUME commands to change the volume status to MASTER or USER status, using the EDGCVTC5 REXX procedure.

EDGJVTC3

- Create CHANGEVOLUME commands to build a chain of volumes, using the EDGCVTC6 REXX procedure.
- Execute all created RMM commands.

Sample REXX Procedure

EDGCVTC0

This exec can be used to generate LISTCAT 'CAT('xxxx') ALL commands.

Input Data Sets

Table 34. EDGCVTC0 Input Data Sets

DDNAME	Description
LISTCAT	Sequential file from LISTCAT UCAT

Output Data Sets

Table 35. EDGCVTC0 Output Data Sets

DDNAME	Description
CCARDS	Control cards for the next IDCAMS LISTC job

Return Codes

Table 36. EDGCVTC0 Return Codes

Return code	Description
0	The function was executed successfully
4	Unable to locate an entry
8	An entry could not be defined
12	The requested function could not be performed
16	Severe error

EDGCVTC4

This exec can be used to create ADDRACK, ADDVOLUME and CHANGEVOLUME commands to build a RMM CDS from Manual Management input.

Input Data Sets

Table 37. EDGCVTC4 Input Data Sets

DDNAME	Description
RMM	RMM extract file created by EDGHSP
CAT	Sequential file created by IDCAMS (LISTCAT)

Output Data Sets

Table 38. EDGCVTC4 Output Data Sets

DDNAME	Description
ERROR	Error messages
TEMP	Temporary file for the next REXX procedure EDGCVTC5

Return Codes

Table 39. EDGCVTC4 Return Codes

Return code	Description
0	The function was executed successfully
4	Unable to locate an entry
8	An entry could not be defined
12	The requested function could not be performed
16	Severe error

EDGCVTC5

Creating CHANGEVOLUME commands for the first dataset on a volume.

Creating ADDDATASET commands if you have more than one file on a volume. When you add a data set, the preceding data set on the volume must already be defined. If the preceding data set is missing, a dummy file named **DUMMY.FILEsequence_number.MISSING** or **DUMMY.LBLsequence_number.MISSING** is added.

If you have the "Duplicate files on volume" condition, please check and correct your input file and rerun the job.

Input Data Sets

Table 40. EDGCVTC5 Input Data Sets

DDNAME	Description
MVMF	Temporary catalog sort file
TEMP	Temporary file created by exec EDGCVTC4

Output Data Sets

Table 41. EDGCVTC5 Output Data Sets

DDNAME	Description
CMDSAR	RMM ADDRACK and ADDVOLUME commands
CMDSAD	RMM ADDDATASET commands
CMDSCV	RMM CHANGEVOLUME commands
ERROR	Error messages

Return Codes

Table 42. EDGCVTC5 Return Codes

Return code	Description
0	The function was executed successfully
4	Unable to locate an entry
8	An entry could not be defined
12	The requested function could not be performed
16	Severe error

EDGCVTC6

This exec can be used to create a CHANGEVOLUME command to build volume chains.

Input Data Sets

Table 43. EDGCVTC6 Input Data Sets

DDNAME	Description
TEMP	Temporary catalog sort file

Output Data Sets

Table 44. EDGCVTC6 Output Data Sets

DDNAME	Description
CMDSCVM	RMM CHANGEVOLUME commands

Return Codes

Table 45. EDGCVTC6 Return Codes

Return code	Description
0	The function was executed successfully
4	Unable to locate an entry
8	An entry could not be defined
12	The requested function could not be performed
16	Severe error

Chapter 6. Conversion from TAPE2000

For general conversion process questions refer to "**Converting to Removable Media Manager A Practical Guide**" (SG24-4998). You can check for the Redbook "**Converting to Removable Media Manager A Practical Guide**" (SG24-4998) on : <http://www.redbooks.ibm.com> and search for "RMM".

If you do not have the redbook we recommend you to order a copy. Until your copy arrives the following summary is included so that you can understand the overall process for conversion of data to DFSMSrmm and can execute the extract and conversion utilities and jobs.

The following steps summarize the recommended conversion process.

1. Review the "**Converting to Removable Media Manager A Practical Guide**" (SG24-4998).
2. Extract data from the TAPE2000 database, creating an extract file.
3. Evaluate existing retention management and volume movement policies.
4. Familiarize yourself with the EDGJT2EX sample conversion job.
5. Formulate EDGCNVT control statements.
6. Define and customize the RMM system.
7. Start RMM subsystem in MANUAL mode to get familiarization with dialog and commands.
8. Execute the EDGJT2EX conversion job and check the messages produced.
9. Execute the EDGCNVT and EDGCEBIN programs.
10. Load the RMM Control Data Set.
11. Use EDGUTIL to verify the RMM control data set, and to create the control record.
12. Start RMM subsystem in RECORD mode.
13. Manually update any VRS definitions and save changes.
14. Execute RMM housekeeping process (VRSEL, EXPROC and DSTORE) to assign storage locations and bin numbers.
15. Compare RMM scratch and storage location reports with corresponding TAPE2000 reports.
16. Correct any conversion problems and repeat steps 8 to 16.
17. Modify and Install any required installation exits, such as EDGUX100.
18. Start RMM in WARNING mode.
19. When no unexplained differences, start RMM in PROTECT mode and remove TAPE2000 from the system.

Following table lists the conversion samples, programs, and JCL for the extraction of data from TAPE2000

Table 46. DFSMSrmm-supplied conversion samples, programs, and JCL

Name	Location	Function
EDGRT2EX	WEB	Source for TAPE2000 extract exec
EDGJT2EX	WEB	JCL to execute EDGRT2EX TAPE2000 conversion exec
EDGCNVT	LINKLIB	The EDGCNVT program
EDGCNXIT	SAMPLIB	Source for sample exit for EDGCNVT
EDGCEBIN	LINKLIB	The EDGCEBIN program
EDGJCNVT	SAMPLIB	JCL to execute EDGCNVT and EDGCEBIN
EDGCVOVL	LINKLIB	The EDGCVOVL program
EDGJLOAD	SAMPLIB	JCL to execute EDGCVOVL, create and load CDS

Table 46. DFSMSrmm-supplied conversion samples, programs, and JCL (continued)

Name	Location	Function
EDGJVERR	SAMPLIB	JCL to create a CDS control record and execute EDGUTIL VERIFY processing

The sample JCL is set up to work in a "typical" installation. Some of the sample JCL is commented out so that you choose to use it or not as is your preference.

Functional Description

The data extraction exec is designed to extract relevant information from the TAPE2000 extract file. The information is used to create a single, sequential output file containing a series of records that are input to the EDGCNVT program before they are loaded into the DFSMSrmm CDS (see Table 47).

The LRECL of the output files is coded in the JCL and also copied from input files. The value included in the JCL examples and samples is just an indication of the approximate record length.

Table 47. EDGRT2EX Output Data Sets and Records

DDNAME	Output Description	Output Record
DEXTOUT	<ul style="list-style-type: none"> • Data sets and volumes • Retention and storage location management policies • VRS information 	<ul style="list-style-type: none"> • D (data set) record • L (volume) record • K (VRS) record

Note that the data set and volume (D-type and L-type) records are sequence dependent - see Chapter 9, "Building the DFSMSrmm CDS," on page 185 for details.

Preparing to run the Extract Program

Before you begin the extraction process you should consider verifying the integrity of the TAPE2000 database to check for broken volume chains, expiration dates in error, or any other condition that can cause an error during data extraction.

You must run the appropriate TAPE2000 utilities to verify the integrity of the database and correct the errors.

Once you are sure that you have a valid database, you can proceed to the next step

Note: The EDGRT2EX exec does include some processing to correct minor errors in the extracted data, but you should use the messages issued by the exec to correct the TAPE2000 data and re-extract valid data.

You use the TAPE2000 TPMRPEXT utility to create an extract, and sort the data into volume sequence. Sample JCL for creating the extract is supplied in the TAPE2000 JCLLIB library as member TPMRPEXT.

```

//EXTR12 EXEC PGM=TPMRPEXT,REGION=4M,PARM='TLFS'
//SYSTRC DD SYSOUT=*
//SYSMSG DD SYSOUT=*
//SCRMSG DD SYSOUT=*
//SCRPRR DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//T2MAST DD DSN=TAPE2000.V623.T2CATLG,
//          DISP=SHR
//REPEXT DD DSN=WOODMW.T2000.EXTRACT.UNSORTED.DATA,
//          DISP=(,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,(50,12),RLSE),
//          DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB)
//SCRIN DD *
PROCEDURE TPMRPEXT
EXT_VAULTS = "Y" /* VAULT TABLE */
EXT_VOLDNS = "Y" /* VOLSER/DATASET INFO */
EXT_PATTERNS = "Y" /* PATTERN MASTER */
EXT_VLTDEF = "Y" /* VAULT DEFINITIONS */
EXT_POOLDEF = "Y" /* POOLING DEFINITIONS */
END
/*
//*****
//* SORT EXTRACT FILE BY 1) VOLSER, 2) PHY SEQ, 3) RECORD TYPE **
//*****
//EXTR13 EXEC PGM=SORT,REGION=4M
//SYSOUT DD SYSOUT=*
//SORTWK01 DD SPACE=(CYL,(15,5)),UNIT=SYSWRK
//SORTWK02 DD SPACE=(CYL,(15,5)),UNIT=SYSWRK
//SORTWK03 DD SPACE=(CYL,(15,5)),UNIT=SYSWRK
//SORTIN DD DSN=WOODMW.T2000.EXTRACT.UNSORTED.DATA,DISP=SHR
//SORTOUT DD DSN=WOODMW.T2000.EXTRACT.DATA,
//          DISP=(NEW,CATLG,DELETE),
//          DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//          SPACE=(CYL,(50,12),RLSE),
//          UNIT=SYSDA
//SYSIN DD *
SORT FIELDS=(7,6,A,17,4,A,5,2,A),EQUALS,FORMAT=BI
END
/*

```

Figure 34. Sample JCL to Execute TPMRPEXT

The EDGRT2EX Exec

This section describes the use and function provided by the EDGRT2EX data extraction exec.

EDGRT2EX supports the latest releases of TAPE2000.

EDGRT2EX provides different functions that must be selected in the correct sequence. These update, correct, and use the information in the extracted data. You request the functions by passing a parameter, which can be the following:

FUNCTION

You use this keyword to request the function, which must be one of the following. The values are listed in the order they must be processed.

CVOL

Requests that the controlling volume information is checked for volumes and set for data sets. The TAPE2000 extracted data set information does not have the controlling volume set.

The input is assumed to be sorted by volume serial number.

FVSEQ

Requests that the family sequence number of the data set records is checked and set for data sets. The TAPE2000 extracted data set information does not have the family sequence number set.

The input must first be sorted into multi-volume sets so that all data sets in the same family are sorted in creation sequence.

VSEQ

Requests that the family sequence number, volume sequence number, and controlling volume for volumes and data sets is checked and corrected. The TAPE2000 extracted volume information may have incorrect volume and family sequence numbers. There are also missing additional data set records in the extract that must be processed.

The input must first be sorted by volume with the most recent data set on a volume first in sequence and the volume last in sequence.

BUILD

Requests that RMM conversion records are built from the corrected TAPE2000 extract data.

The input must first be sorted into family sets in volume sequence and file sequence order.

OPT You use the keyword to pass additional options to some functions. The supported options are:

MVOL

Use this with the BUILD function to request that multi-volume sequence checking is performed. We recommend that you use this option.

Input Files

The input file required depends on the function you request.

Table 48. EDGRT2EX Input Files

FUNCTION	DDNAME	Description
CVOL	T2RPTEXT	Contains the sorted, extracted TAPE2000 data
FVSEQ	DATASETS	Contains just the extract data set records sorted by family
VSEQ	MERGED	Contains all records from the extract sorted by volume.
BUILD	T2RPTEXT	Contains updated extract data sorted by family set and volume and file sequence in family.

Output Files

The output file required depends on the function you request.

Table 49. EDGRT2EX Output Files

FUNCTION	DDNAME	Description
ANY	SYSTSPRT	Contains all messages issued by EDGRT2EX.
CVOL	UPDATED	Receives the updated extract records.
FVSEQ	SEQFIXED	Receives the updated extract data set records.
VSEQ	VSEQFIX	Receives all records including those updated by EDGRT2EX
BUILD	DEXTOUT	Receives all records build by EDGRT2EX ready for input to EDGCNVT.

Converted Data Fields

EDGRT2EX converts as much data as is possible to RMM. Sometimes the exact data needed in RMM is not available, so either we set default values or use similar existing data from TAPE2000.

The volume and data set owner is set from the creating jobname of the first file on the volume.

The volume medianame is set to either CARTS or REELS depending on the type of volume defined to TAPE2000. To use different values you can use SYSIN statements in EDGCNVT processing.

No expiration dates are available in the extract so you must ensure that under RMM all existing data is retained or managed by a VRS.

Execution

Following figure shows the sample JCL to execute the extract process for TAPE2000. See sample EDGJT2EX for the supplied sample JCL.

```
//CLEAN EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE 'WOODMW.T2000.UPDATED.DATA' PURGE
DELETE 'WOODMW.T2000.DATASETS.DATA' PURGE
DELETE 'WOODMW.T2000.THEREST.DATA' PURGE
DELETE 'WOODMW.T2000.SORTEDDS.DATA' PURGE
DELETE 'WOODMW.T2000.SEQFIXED.DATA' PURGE
DELETE 'WOODMW.T2000.MERGED.DATA' PURGE
DELETE 'WOODMW.T2000.VSEQFIX.DATA' PURGE
DELETE 'WOODMW.T2000.FINAL.DATA' PURGE
DELETE 'WOODMW.T2000.EDGRT2EX.DEXTOUT.DATA' PURGE
SET MAXCC = 0
//*
```

Figure 35. (Part 1 of 5). Sample JCL to Execute EDGRT2EX

```
//CVOL EXEC PGM=IKJEFT01,REGION=32M
//SYSPRT DD SYSOUT=*
//SYSTSIN DD *
EXEC 'WOODMW.T2CNV.EXEC(EDGRT2EX)' 'FUNCTION(CVOL)'
//UPDATED DD DISP=(,CATLG),UNIT=SYSALLDA,LRECL=1200,
// RECFM=VB,SPACE=(700,(7000,2000),RLSE),
// DSN=WOODMW.T2000.UPDATED.DATA
//T2RPTEXT DD DISP=SHR,
// DSN=WOODMW.T2000.EXTRACT.DATA
//*
```

Figure 36. (Part 2 of 5). Sample JCL to Execute EDGRT2EX

```

//SORT1 EXEC PGM=ICETOOL,REGION=24M
//TOOLMSG DD SYSOUT=* ICETOOL MESSAGES
//DFSMSG DD SYSOUT=* DFSORT MESSAGES
//*
//UPDATED DD DISP=SHR,DSN=WOODMW.T2000.UPDATED.DATA
//DATASETS DD DISP=(,CATLG),UNIT=SYSDA,
//          DSN=WOODMW.T2000.DATASETS.DATA,
//          SPACE=(CYL,(50,20),RLSE)
//SORTEDDS DD DISP=(,CATLG),UNIT=SYSDA,
//          DSN=WOODMW.T2000.SORTEDDS.DATA,
//          SPACE=(CYL,(50,20),RLSE)
//THEREST DD DISP=(,CATLG),UNIT=SYSDA,
//          DSN=WOODMW.T2000.THEREST.DATA,
//          SPACE=(CYL,(50,20),RLSE)
//TOOLIN DD *
MODE CONTINUE
COPY FROM(UPDATED) USING(SPLT)
SORT FROM(DATASETS) TO(SORTEDDS) USING(SORT)
//SPLTCNTL DD *
OUTFIL FNAMES=DATASETS,
INCLUDE=(5,2,CH,EQ,C'20')
OUTFIL FNAMES=THEREST,
INCLUDE=(5,2,CH,NE,C'20')
//SORTCNTL DD *
OPTION VLSHRT,NOCHALT
*          TYPE = 20 = data sets
INCLUDE COND=(5,2,CH,EQ,C'20')
*          CVOL          CRDATE          CRTIME          LOGSEQ
SORT FIELDS=(21,6,CH,A,77,4,PD,A,81,5,PD,A,117,2,BI,A,
*          VOLSEQ
*          13,4,CH,A)
//*

```

Figure 37. (Part 3 of 5). Sample JCL to Execute EDGRT2EX

```

//FVSEQ   EXEC PGM=IKJEFT01,REGION=32M
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
EXEC 'WOODMW.T2CNV.EXEC(EDGRT2EX)' 'FUNCTION(FVSEQ)'
//SEQFIXED DD DISP=(,CATLG),UNIT=SYSALLDA,LRECL=1200,
//          RECFM=VB,SPACE=(700,(7000,2000),RLSE),
//          DSN=WOODMW.T2000.SEQFIXED.DATA
//DATASETS DD DISP=SHR,
//          DSN=WOODMW.T2000.SORTEDDS.DATA
//*
//SORT2   EXEC PGM=ICETOOL,REGION=24M
//TOOLMSG DD SYSOUT=*   ICETOOL MESSAGES
//DFSMSG  DD SYSOUT=*   DFSORT  MESSAGES
//*
//ALL     DD DISP=SHR,DSN=WOODMW.T2000.SEQFIXED.DATA
//        DD DISP=SHR,DSN=WOODMW.T2000.THEREST.DATA
//MERGED  DD DISP=(,CATLG),UNIT=SYSDA,
//        DSN=WOODMW.T2000.MERGED.DATA,
//        SPACE=(CYL,(50,20),RLSE)
//TOOLIN DD *
MODE CONTINUE
SORT FROM(ALL) TO(MERGED) USING(SORT)
//SORTCNTL DD *
OPTION VLSHRT,NOCHALT
*          VOL          TYPE          CRDATE          CRTIME
SORT  FIELDS=(7,6,CH,A,5,2,CH,D,77,4,PD,D,81,5,PD,D)
//VSEQ    EXEC PGM=IKJEFT01,REGION=32M
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
EXEC 'WOODMW.T2CNV.EXEC(EDGRT2EX)' 'FUNCTION(VSEQ)'
//VSEQFIX DD DISP=(,CATLG),UNIT=SYSALLDA,LRECL=1200,
//        RECFM=VB,SPACE=(700,(7000,2000),RLSE),
//        DSN=WOODMW.T2000.VSEQFIX.DATA
//MERGED  DD DISP=SHR,
//        DSN=WOODMW.T2000.MERGED.DATA
//*

```

Figure 38. (Part 4 of 5). Sample JCL to Execute EDGRT2EX

```

//SORT3 EXEC PGM=ICETOOL,REGION=24M
//TOOLMSG DD SYSOUT=* ICETOOL MESSAGES
//DFSMSG DD SYSOUT=* DFSORT MESSAGES
//*
//VSEQFIX DD DISP=SHR,DSN=WOODMW.T2000.VSEQFIX.DATA
//FINAL DD DISP=(,CATLG),UNIT=SYSDA,
// DSN=WOODMW.T2000.FINAL.DATA,
// SPACE=(CYL,(50,20),RLSE)
//TOOLIN DD *
MODE CONTINUE
SORT FROM(VSEQFIX) TO(FINAL) USING(SORT)
//SORTCNTL DD *
OPTION VLSHRT,NOCHALT
* CVOL FAMSEQ FILESEQ TYPE
SORT FIELDS=(21,6,CH,A,27,4,CH,A,17,4,CH,A,5,2,CH,A)
//*
//BUILD EXEC PGM=IKJEFT01,REGION=32M
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
EXEC 'WOODMW.T2CNV.EXEC(EDGRT2EX)' 'FUNCTION(BUILD) OPT(MVOL) '
//DXTOUT DD DISP=(,CATLG),UNIT=SYSALLDA,LRECL=1200,
// RECFM=VB,SPACE=(700,(7000,2000),RLSE),
// DSN=WOODMW.T2000.EDGRT2EX.DXTOUT.DATA
//T2RPTEXT DD DISP=SHR,
// DSN=WOODMW.T2000.FINAL.DATA

```

Figure 39. (Part 5 of 5). Sample JCL to Execute EDGRT2EX

Return Codes

Table 50. EDGRT2EX Return Codes

Return code	Description
0	Normal completion. No errors or warnings detected.
4	One or more warning messages issued. Valid records for input to EDGCNVT are produced, but the warning messages should be reviewed to ensure that results are as expected.
8	One or more error messages issued. Unless you correct the reported error you will not get expected results under RMM. The errors should be corrected and the conversion process rerun.
12	An error has been encountered in the exec processing. Refer to the REXX error messages for resolution. If the problem is not caused by bad data in the extract file, report the error to IBM.

Messages

EDGRT2EX-00000 VOLUMES UPDATED WITH CONTROL VOLUME = *count* 'RC=04'

Problem determination: where

count Missing volumes count.

Severity: Warning

Explanation: This is the count of found volumes without controlling the volume name set in the extract header.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00005 POSSIBLE OLD DATA SETS = count 'RC=04'

Problem determination: where

count Possible old data sets count

Severity: Warning

Explanation: This is count of founded data sets which possible are old.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00010 DATA SETS WITH CORRECTED FAMILY SEQUENCE = count 'RC=04'

Problem determination: where

count Count of data sets with corrected family sequence.

Severity: Warning

Explanation: This is count of data sets with corrected family sequence.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00015 DATA SETS WITH CORRECTED CONTROL VOLUME = count 'RC=04'

Problem determination: where

count Count of data sets with corrected control volume.

Severity: Warning

Explanation: This is count of data sets with corrected control volume.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00020 VOLUMES WITH CORRECTED CONTROL VOLUME = count 'RC=04'

Problem determination: where

count Count of volumes with corrected control volume.

Severity: Warning

Explanation: This is count of volumes with corrected control volume.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00025 VOLUMES WITH CORRECTED FAMILY SEQUENCE = count 'RC=04'

Problem determination: where

count Count of volumes with corrected family sequence.

Severity: Warning

Explanation: This is count of volumes with corrected control volume.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00030 VOLUMES WITH CORRECTED VOLUME SEQUENCE = *count* 'RC=04'

Problem determination: where

count Count of volumes with corrected volume sequence.

Severity: Warning

Explanation: This is count of volumes with corrected volume sequence.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00035 MULTI-VOLUME CHAINS WITH SEQUENCE ERRORS = *count* 'RC=04'

Problem determination: where

count Count of multi-volume chains with sequence errors.

Severity: Warning

Explanation: This is count of multi-volume chains with sequence errors.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00040 VOLUMES CREATED FOR MISSING VOLUMES = *count* 'RC=04'

Problem determination: where

count Count of volumes created for missing volumes.

Severity: Information

Explanation: This is count of volumes created for missing volumes.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00045 VOLUMES WITH FILE 1 MISSING = *count* 'RC=04'

Problem determination: where

count Count of volumes with file 1 missing.

Severity: Warning

Explanation: This is count of volumes with file 1 missing.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00045 DATA SETS OUT OF SEQUENCE AND SKIPPED = *count* 'RC=04'

Problem determination: where

count Count of data sets out of sequence and skipped.

Severity: Warning

Explanation: This is count of data sets out of sequence and skipped.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00050 EXTRA DATA SETS SKIPPED = count

Problem determination: where

count Count of extra data sets skipped.

Severity: Informational

Explanation: This is count of extra data sets skipped.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00055 VOLUMES CREATED DESPITE MISSING DATA SETS = count 'RC=04'

Problem determination: where

count Count of volumes created despite missing data sets.

Severity: Warning

Explanation: This is count of volumes created despite missing data sets.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00060 VOLUMES CREATED DESPITE MISSING DATA SETS = count 'RC=04'

Problem determination: where

count Count of volumes created despite missing data sets.

Severity: Warning

Explanation: This is count of volumes created despite missing data sets.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00065 VOLUMES WITH CORRECTED DATA SET COUNT = count 'RC=04'

Problem determination: where

count Count of volumes with corrected data set count.

Severity: Warning

Explanation: This is count of volumes with corrected data set count.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00070 VAULT DEFINITIONS WITH UNSUPPORTED TYPE = count 'RC=04'

Problem determination: where

count Count of vault definitions with unsupported type.

Severity: Warning

Explanation: This is count of vault definitions with unsupported type.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00075 VOLUMES WITH FILE 1 NOT MATCHING = *count* 'RC=04'

Problem determination: where

count Count of volumes with file 1 not matching.

Severity: Warning

Explanation: This is count of volumes with file 1 not matching.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00080 UNSUPPORTED FUNCTION *function*

Problem determination: where

function Function requested.

Severity: Warning

Explanation: Unsupported function requested.

System action: The job abnormally ends.

User response: Correct JCL and rerun the job.

EDGRT2EX-00085 NO CONTROLLING VOLUME FOR *volser* - SETTING FROM VOLSER

Problem determination: where

volser Volume serial number.

Severity: Warning

Explanation: A volume was found to have no controlling volume name set in the extract header.

System action: Processing continues and sets the controlling volume to the volume serial number.

User response: Use the information from TAPE2000 to determine if the volume should have a controlling volume, and correct it if required.

Alternatively, if the information converted by EDGRT2EX is acceptable after you have checked it, you can ignore this error.

EDGRT2EX-00090 CORRECTING FAMSEQ FOR DATA SET ON VOLUME *volser* CONTROL VOLUME *cool* FILE *filenum* DSNAME *dsname*

Problem determination: where

volser Volume serial number.

cool Control volume serial number.

filenum File sequence number.

dsname Data set name.

Severity: Warning

Explanation: The family sequence and controlling volume information from the most recently created file on a volume is being used to check and correct the information for other files on the volume and also check and correct the volume information. The data set listed is being corrected.

System action: Processing continues and sets the family sequence number.

User response: Processing by FUNCTION(VSEQ) stage may be changing information changed by the FUNCTION(FVSEQ) or may be correcting incorrect information in TAPE2000 data. Also review the messages issued by FUNCTION(FVSEQ) and FUNCTION(BUILD) processing for this volume and controlling volume to see what processing has been performed and what corrections made.

One reason for this message is that the data set is old; they may also be listed in message FF01, and BR04 or BR05.

Use the information from TAPE2000 to determine if the correct sequence has been determined. Perhaps there is missing or incorrect information in the TAPE2000 data and it can be corrected manually.

Alternatively, if the information converted by EDGRT2EX is acceptable after you have checked it, you can ignore this error.

EDGRT2EX-00095 CORRECTING CONTROL VOLUME FOR DATA SET ON VOLUME *volser* CONTROL VOLUME *cvol* FILE *filenum* DSNAME *dsname*

Problem determination: where

volser Volume serial number.

cvol Control volume serial number.

filenum File sequence number.

dsname Data set name.

Severity: Warning

Explanation: The family sequence and controlling volume information from the most recently created file on a volume is being used to check and correct the information for other files on the volume and also check and correct the volume information. The data set listed is being corrected.

System action: Processing continues and sets the controlling volume.

User response: Processing by FUNCTION(VSEQ) stage may be changing information changed by the FUNCTION(FVSEQ) or may be correcting incorrect information in TAPE2000 data. Also review the messages issued by FUNCTION(FVSEQ) and FUNCTION(BUILD) processing for this volume and controlling volume to see what processing has been performed and what corrections made.

Use the information from TAPE2000 to determine if the correct controlling volume has been determined. Perhaps there is missing or incorrect information in the TAPE2000 data and it can be corrected manually.

Alternatively, if the information converted by EDGRT2EX is acceptable after you have checked it, you can ignore this error.

EDGRT2EX-00100 CORRECTING CONTROL VOLUME FOR VOLUME *volser* FROM *cvol* TO *newcvol*

Problem determination: where

volser Volume serial number.

cvol Old control volume serial number.

newcvol New control volume serial number.

Severity: Warning

Explanation: The family sequence and controlling volume information from the most recently created file on a volume is being used to check and correct the information for other files on the volume and also check and correct the volume information. The volume listed is being corrected.

System action: Processing continues and sets the controlling volume.

User response: Processing by FUNCTION(VSEQ) stage may be changing information changed by the FUNCTION(FVSEQ) or may be correcting incorrect information in TAPE2000 data. Also review the messages issued by FUNCTION(FVSEQ) and FUNCTION(BUILD) processing for this volume and controlling volume to see what processing has been performed and what corrections made.

Use the information from TAPE2000 to determine if the correct controlling volume has been determined. Perhaps there is missing or incorrect information in the TAPE2000 data and it can be corrected manually.

Alternatively, if the information converted by EDGRT2EX is acceptable after you have checked it, you can ignore this error.

EDGRT2EX-00105 CORRECTING FAMSEQ ON VOLUME *volser* CONTROL VOLUME *cool*

Problem determination: where

volser Volume serial number.

cool Control volume serial number.

Severity: Warning

Explanation: The family sequence and controlling volume information from the most recently created file on a volume is being used to check and correct the information for other files on the volume and also check and correct the volume information. The volume listed is being corrected.

System action: Processing continues and sets the family sequence number.

User response: Processing by FUNCTION(VSEQ) stage may be changing information changed by the FUNCTION(FVSEQ) or may be correcting incorrect information in TAPE2000 data. Also review the messages issued by FUNCTION(FVSEQ) and FUNCTION(BUILD) processing for this volume and controlling volume to see what processing has been performed and what corrections made.

Use the information from TAPE2000 to determine if the correct sequence has been determined. Perhaps there is missing or incorrect information in the TAPE2000 data and it can be corrected manually.

Alternatively, if the information converted by EDGRT2EX is acceptable after you have checked it, you can ignore this error.

EDGRT2EX-00110 CORRECTING VOLSEQ ON VOLUME *volser* CONTROL VOLUME *cool*

Problem determination: where

volser Volume serial number.

cool Control volume serial number.

Severity: Warning

Explanation: The family sequence and controlling volume information from the most recently created file on a volume is being used to check and correct the information for other files on the volume and also check and correct the volume information. The volume listed is being corrected.

System action: Processing continues and sets the volume sequence number.

User response: Processing by FUNCTION(VSEQ) stage may be changing information changed by the FUNCTION(FVSEQ) or may be correcting incorrect information in TAPE2000 data. Also review the messages issued by FUNCTION(FVSEQ) and FUNCTION(BUILD) processing for this volume and controlling volume to see what processing has been performed and what corrections made.

Use the information from TAPE2000 to determine if the correct sequence has been determined. Perhaps there is missing or incorrect information in the TAPE2000 data and it can be corrected manually.

Alternatively, if the information converted by EDGRT2EX is acceptable after you have checked it, you can ignore this error.

EDGRT2EX-00115 POSSIBLE OLD DATA SET ON VOLUME *volser* CONTROL VOLUME *cool* FILE *filenum* DATA SET *dsname*

Problem determination: where

volser Volume serial number.

cool Control volume serial number.

filenum File sequence number.

dsname Data set name.

Severity: Warning

Explanation: Input to this stage is a list of files sorted by controlling volume and creation order. The first record for

a family is expected to be file 1 on the controlling volume for that family. The data set listed has been found before the first file for the controlling volume.

System action: Processing continues and sets the controlling volume to the volume serial number, and the family sequence number to 1.

User response: Processing by the FUNCTION(VSEQ) stage may again change the controlling volume or family sequence number and be able to automatically overcome this error. Also review the messages issued by FUNCTION(VSEQ) and FUNCTION(BUILD) processing for this volume and controlling volume to see what processing has been performed and what corrections made.

By looking at the records in the intermediate file input to this stage you can see the results of sorting the records. Use the information from TAPE2000 to determine if the correct sequence has been determined. Perhaps there is missing or incorrect information in the TAPE2000 data.

Suggested action is to remove the data sets from TAPE2000 because they are old records no longer required since the volume was reused.

Alternatively, if the information converted by EDGRT2EX is acceptable after you have checked it, you can ignore this error.

EDGRT2EX-00130 FAMILY SEQUENCE MATCH FOR VOLUME *volser* AND VOLUME *prevvol* CONTROL VOLUME *cvol*

Problem determination: where

volser Volume serial number.

prevvol Previous volume serial number.

cvol Control volume serial number.

Severity: Warning

Explanation: Two volumes in the same family have the same family volume sequence number. The most likely cause of this message is a missing data set record for the *prevvol*.

System action: Processing continues using the same sequence number for both volumes.

User response: Use the information from TAPE2000 to determine the correct volume sequence and correct the information in TAPE2000 before rerunning the extract and the conversion. There may also be messages from earlier stages in the EDGRT2EX processing for this volume or family of volumes that help identify the possible error.

Suggested action is to check the *prevvol* volume. If it was perhaps rejected at the original creation time and has no data set on the volume you could scratch the volume.

EDGRT2EX-00135 VOLUME RECORD MISSING FOR *volser* - CREATING VOLUME RECORD SCRATCH(*y/n*). DATA SET *dsname*

Problem determination: where

volser Volume serial number.

y/n Scratch volume record created or not.

dsname Data set name.

Severity: Warning

Explanation: A data set record was found in the extract file with no matching volume record.

System action: Processing continues. A L record is created for the missing volume using the status information for the first data set on the volume. Some volume information is modelled on the previous volume processed.

User response: Use the information from TAPE2000 to determine why the volume is missing. Either add the volume to TAPE2000 or delete the data set information from TAPE2000 if it is no longer valid.

Alternatively, if the information converted by EDGRT2EX is acceptable after you have checked it, you can ignore this error.

EDGRT2EX-00140 FILE 1 MISSING FOR VOLUME *volser* **VOLDSN1** *dsnameforfile1* **VOLDSN1CRDATE/TIME**
date/time

Problem determination: where

volser Volume serial number.

dsnameforfile1

First file data set name.

date/time

First file data set creation date and time.

Severity: Warning

Explanation: A data set record for the first physical file on a volume is missing and there are files greater than file 1 on the volume

System action: Processing continues. The next data set in sequence on the volume is treated as the first file on the volume.

User response: Use the information from TAPE2000 to determine why the data set record is missing. Add the data set to TAPE2000 or cleanup the other data set information for the volume if it is no longer valid.

Alternatively, if the information converted by EDGRT2EX is acceptable after you have checked it, you can ignore this error.

EDGRT2EX-00145 DATA SET OUT OF SEQUENCE FOR VOLUME *volser* - **SKIPPING** *current/previous* **RECORD**
FILE *fnum* **DATA SET** *dsname*

Problem determination: where

volser Volume serial number.

current/previous

Current or previous volume serial number.

fnum File sequence number.

dsname Data set name.

Severity: Warning

Explanation: Two data set records for the same physical file on a volume exist. One of the files is no longer valid because it has probably been overwritten.

System action: Processing continues by skipping one of the files. The newest file is selected based on the creating data and time.

User response: Use the information from TAPE2000 to determine why the old data set record is present. Cleanup the old data set information for the volume if it is no longer valid.

Alternatively, if the information converted by EDGRT2EX is acceptable after you have checked it, you can ignore this error.

EDGRT2EX-00150 EXTRA DATA SET FOUND FOR VOLUME *volser* - **SKIPPING** **RECORD.** **FILE** *fnum* **DATA SET**
dsname

Problem determination: where

volser Volume serial number.

fnum File sequence number.

dsname Data set name.

Severity: Information

Explanation: According to TAPE2000 data, the named file is surplus to requirements. The file number is greater than the TAPE2000 recorded number of files for the volume.

System action: Processing continues by skipping the file.

User response: Use the information from TAPE2000 to determine why the old data set record is present. Cleanup the old data set information for the volume if it is no longer valid.

Alternatively, if the information converted by EDGRT2EX is acceptable after you have checked it, you can ignore this error.

**EDGRT2EX-00155 MISSING DATA SET ON VOLUME *volser* - CREATING VOLUME RECORD. SCRATCH(Y)
VOLDSN1 *dsnameforfile1* VOLDSN1CRDATE/TIME *date/time***

Problem determination: where

volser Volume serial number.

dsnameforfile1

First file data set name.

date/time

First file data set creation date and time.

Severity: Warning

Explanation: A volume record in the extract file specifies that the recorded number of files for the volume is greater than zero, but no data set records exist for the volume.

System action: Processing continues by creating the volume without a data set name and the file count set to zero.

User response: Use the information from TAPE2000 to determine why the data set record is missing. Add the data set information to TAPE2000 using the information in the volume record, or correct the number of data sets on the volume.

Alternatively, if the information converted by EDGRT2EX is acceptable after you have checked it, you can ignore this error.

EDGRT2EX-00160 NUMBER OF DATA SETS *dsnno* ON VOLUME *volser* CORRECTED TO *filecount* - NOW MATCHES THE NUMBER OF VALID RECORDS FOUND

Problem determination: where

dsnno Number of data sets on volume.

volser Volume serial number.

filecount

Corrected number of data sets on volume.

Severity: Warning

Explanation: A volume record in the extract file specifies that the recorded number of files for the volume is different than the number of data set records found to be valid for the volume.

System action: Processing continues by updating the count of files on the volume.

User response: Use the information from TAPE2000 to determine why the count is incorrect, and correct it if possible.

Alternatively, if the information converted by EDGRT2EX is acceptable after you have checked it, you can ignore this error.

EDGRT2EX-00165 RETENTION TYPE *type* NOT SUPPORTED CONVERTED TO PERMANENT.

Problem determination: where

type Retention type.

Severity: Error

Explanation: A vault definition includes a vaulting retention type that is not supported by RMM.

System action: Processing continues by setting the retention type to LASTREFERENCEDAYS and the time limit to permanent.

User response: Review your use of this retention type and compare it to those available in RMM. If there is an equivalent in RMM either update the vaulting definition to use a type that can be converted to RMM or edit the VRSCMNDS file that is created by EDGCNVT to a value near equivalent to RMM, or use the RMM dialog to change the VRS created by conversion once you have started RMM.

EDGRT2EX-00170 FILE 1 DOES NOT MATCH VOLUME *volser* **VOLDSN1** *dsnameforfile1* **DATA SET DSN** *dsname*
VOLDSN1CRDATE/TIME *date1/time1* **DSNCRDATE/TIME** *date2/time2*

Problem determination: where

volser Volume serial number.

dsnameforfile1

First file data set name.

dsname Data set name.

date1/time1

First file data set creation date and time.

date2/time2

Data set creation date and time.

Severity: Warning

Explanation: A volume record contains information for the first file on the volume, but the first data set record for file 1 does not match this information. The creation date and time are used to trigger this message.

System action: Processing continues by using the data set record found as the first file and ignoring the volume file 1 information.

User response: Use the information from TAPE2000 to determine why the data set record for file 1 is missing or different from that named in the volume record. Correct the data set information in TAPE2000 using the information in the volume record, or cleanup the old data set records for the volume.

Alternatively, if the information converted by EDGRT2EX is acceptable after you have checked it, you can ignore this error.

EDGRT2EX-00175 SCRATCH VOLUMES UPDATED WITH CONTROL VOL = *count* **'RC=04'**

Problem determination: where

count Missing volumes count.

Severity: Warning

Explanation: This is the count of found scratch volumes without the controlling volume name set in the extract header.

System action: The utility ends.

User response: Refer to the accompanying messages for the appropriate action.

EDGRT2EX-00180 WRONG CONTROLLING VOLUME FOR SCRATCH *volser* **- SETTING FROM VOLSER**

Problem determination: where

volser Volume serial number.

Severity: Warning

Explanation: Wrong controlling volume was found for scratch volume.

System action: Processing continues and sets the controlling volume to the volume serial number.

User response: Use the information from TAPE2000 to determine why the volume has a wrong controlling volume, and correct it if required.

Summary Messages

Each time the EDGRT2EX exec runs it can issue multiple error/warning messages, but it also issues a summary of processing such as that in the following:

Number of extract records read	=	40992	
Number of volume records read	=	6283	
Number of data set records read	=	33896	
Number of Vault records read	=	17	
Number of Pattern definitions read	=	741	
Number of Vault definitions read	=	56	
Number of Lrecords created	=	6338	
Number of Drecords created	=	27480	
Number of Kdsn records created	=	741	
Number of Kname records created	=	105	
EDGRT2EX-00040 Volumes created for missing volumes	=	56	
EDGRT2EX-00060 Volumes created despite missing data sets	=	29	RC=04
EDGRT2EX-00045 Volumes with file 1 missing	=	7	RC=04
EDGRT2EX-00075 Volumes with file 1 not matching	=	5	RC=04
EDGRT2EX-00050 Data sets out of sequence and skipped	=	91	RC=04
EDGRT2EX-00055 Extra data sets skipped	=	60	
EDGRT2EX-00065 Volumes with corrected data set count	=	12	RC=04
EDGRT2EX-00070 Vault definitions with unsupported type	=	0	RC=08
EDGRT2EX-00035 Multi-volume chains with sequence errors	=	1	RC=04
Highest return code is 04			

Figure 40. EDGRT2EX Run Summary Messages example

The first set of summary messages show which records were found and processed.

The second set, if they are produced, show what records are created by the BUILD function.

The final set of messages summarize by each of the possible errors. The message number is displayed along with a concise statement of the error and a count of records with that problem and the return code value set for that message. Use the message number to search for the messages issued for the records that are affected. Review other errors reported for the same volume or family of volumes as some errors may be interrelated. Use the message explanations and user actions to decide what to do.

Finally the return code is listed. The return code is the highest of any error detected during processing.

TAPE2000 Conversion Considerations

This section describes the usage considerations for the TAPE2000 to RMM conversion tools.

Storage Locations

RMM supports an almost unlimited number of storage and loan locations. You should be able to convert your vaults to DFSMSrmm without hitting a limit caused by their number. In DFSMSrmm there are three built-in storage locations which are all managed by bin number (slot), but by using LOCDEF commands in the DFSMSrmm parmlib you can use installation defined names, including those you are familiar with under TAPE2000, and decide whether to have them shelf managed or not.

When you convert to DFSMSrmm you have the choice to stay using the vault names you currently use with TAPE2000, or of using different names. When you decide to use different names under DFSMSrmm, you can translate the TAPE2000 vault names to DFSMSrmm storage location names during the EDGCNVT run. To translate the names with EDGCNVT use the LOCDEF SYSIN statement and the IF STORLOC statement to specify a different name.

RMM also supports an almost unlimited number of loan locations. If you wish to use these as a replacement for any of the TAPE2000 vaults, you can do so by using the LOCDEF statement in EDGCNVT to identify the location as a type of LOAN. The EDGRT2EX exec detects loan locations as the names used by volumes being marked out of area, and marks these volumes as being on loan.

TAPE2000 uses the location 'DC' as the 'data center'. In RMM this is most often defined as location SHELF. The EDGRT2EX exec converts DC to SHELF in some cases, but you should include the following in the EDGCNVT SYSIN, to convert DC and DC-R to SHELF:

```
IF STORLOC EQUALS DC      THEN RMMSTORE EQUALS SHELF
IF STORLOC EQUALS DC-R   THEN RMMSTORE EQUALS SHELF
```

Shelf Management

Volumes defined to DFSMSrmm optionally have a rack number assigned to provide shelf management while in the data center, and to provide a unique external identifier for the volume. EDGRT2EX and EDGCNVT normally use the TAPE2000 volume serial number as the DFSMSrmm rack number.

However, if TAPE2000 has a different VOL1 volser for a volume than the volume serial number, EDGRT2EX creates an L record with a different volser and rack number. The VOL1 volser is used as the volume and the volume serial number as the rack number.

Shelf management in DFSMSrmm storage locations is optional, but if provided it is done using bin numbers. Each bin number represents a single slot to store a volume. The EDGRT2EX exec attempts to use the slot numbers for vaults as the bin numbers in RMM. So far, testing with the conversion tools we have found duplicate slot numbers.

Slot numbers assigned to multiple volumes assigned to the same vault. You can use the EDGCNVT SYSIN statements to have the slot numbers ignored by RMM until the duplicates can be resolved.

The EDGRT2EX exec does not detect the duplicate slot numbers. You will find them when you run EDGJLOAD to copy the converted records into the CDS. If there are multiple bin records with the same key, the load will fail.

Note:

EDGRT2EX only converts slot information for slots which are used. If you have gaps in the slot numbers or empty slots available which are not currently used you must add these empty slots to RMM using the ADDBIN command after you have started RMM.

The same approach is required for data center shelf space. EDGRT2EX only converts volumes defined to TAPE2000. Any volumes not currently defined to TAPE2000 or gaps in volser ranges cannot be converted and you must add empty rack numbers and scratch volumes after conversion.

Retention Criteria

The data set and job patterns in TAPE2000 are converted direct to RMM VRSs, using the data set names and job names as masks. The generic masks are converted to those compatible with RMM; * at the end of a mask is usually converted to *.* , and '?' is converted to '%'. No other masks characters are currently supported.

If there is retention information in a pattern this is converted directly into the data set name VRS.

When no retention information is provided, but a vault definition is named the data set name VRS is built using the first vault location retention and the NEXTVRS is set to chain to the second vault information, if any.

Vault definitions are converted to NAME VRS using the Vault definition ID as the NAME value. The first through sixth vaulting values and types are built into NAME VRSs and chained together using names which are simply calculated in numeric sequence.

All retention types are supported except for the 'I' vaulting definition retention type.

There may be a difference in the way RMM and TAPE2000 determine when a data set should no longer be retained by a policy based on a number of days; RMM drops a data set from retention on the calculated date, but TAPE2000 may retain a data set until the day after the calculated date. If this is correct, the numbers of days in a VRS should be increased by 1 day so that the results are the same for both products.

GDG and Non-GDG Processing

No special considerations are included for GDGs. If you have non-generic data set name masks that are to retain GDG data sets you must change the VRSs after conversion or use the VRSCMNDS file generated by EDGCNVT as a base for editing in the VRSs you require. An alternative to editing the VRSCMNDS file is to edit the type 31 extract records to convert non-generic masks to generic masks by adding a '.*' to the end of the data set name. RMM will then consider GDGs for retention by these masks once converted.

Special Expiration Dates

There is no special handling of expiration dates by the EDGRT2EX conversion exec. It is not known what format any 'special dates' may have in TAPE2000 at this time. Existing expiration dates are converted unchanged into RMM. It is the installation's responsibility to ensure that:

1. any new retention criteria are defined as VRS management values and VRS definitions. AND
2. either an appropriate DFSMS ACS routine or RMM installation exit EDGUX100 are installed and active to set VRS management values from special expiration dates.

Multi-Data Set Multi-Volume Retention

DFSMSrmm can return a single volume in a multi-volume set of volumes to scratch - it does not need to operate always on the complete set of volumes. Consequently, You may see DFSMSrmm return a volume to scratch that is still retained by TAPE2000.

Each data set in a multi-volume set can be retained individually using VRS. If there are data sets in the volume set that have different retention requirements DFSMSrmm can attempt to release them earlier. For a volume to be released ALL data sets on that volume must no longer be retained by VRS.

If you use a controlling file to cause the complete set to be retained you may require additional VRS definitions under DFSMSrmm to cause the same volumes to be retained. DFSMSrmm includes parmlib options to ensure that multi-volume sets can be retained as a set or family even when only a single data set is VRS retained. Use the MOVEBY(SET) and RETAINBY(SET) options to manage multi-volume sets as an aggregate or family. Alternatively you can benefit from DFSMSrmm releasing volumes early when the data is not required to be retained.

Parallel Running

UNCATALOG Option

During parallel running, it is strongly recommended that the RMM initialization parameter UNCATALOG in the EDGRMMxx PARMLIB member be set to N. This is especially so if the TAPE2000 retention default is WHILECATALOG. Otherwise, differences in VRS and vault definitions could result in RMM releasing and uncatolog volumes which would normally have been kept by TAPE2000.

Chapter 7. Conversion from Control-M/Tape

This chapter contains information about converting from Control-M/Tape (Control-T), a tape management program of BMC Software, to DFSMSrmm. For general conversion process questions refer to the Redbook **"Converting to DFSMSrmm from Control-T"** (SG24-6243).

If you do not have the Redbook **"Converting to DFSMSrmm from Control-T"** (SG24-6243), we recommend you to order a copy. You can check for the Redbooks on <http://www.redbooks.ibm.com> and search for 'RMM'.

The following steps summarize the recommended conversion process.

1. Review the **"Converting to DFSMSrmm from Control-T"** (SG24-6243), the rest of this chapter, and Chapter 9, "Building the DFSMSrmm CDS," on page 185.
2. Extract data from the Control-M/Tape database, creating an extract file.
3. Familiarize yourself with the sample conversion jobs.
4. Define and customize the DFSMSrmm system.
5. Start DFSMSrmm subsystem in MANUAL mode to get familiarization with dialog and commands.
6. Run EDGJCTT1, EDGJCTT2 and EDGJCTT3 Control-M/Tape conversion jobs to produce a valid input to the EDGCNVT program and check the messages produced.
7. Customize created EDGCNVT SYSIN control statements (see "SYSIN Control Statements" on page 191) and run EDGJCNVT job (see "Running EDGCNVT and EDGCEBIN" on page 187).
8. Run EDGJLOAD job to chain owner records to volume records and load the DFSMSrmm CDS (see "Run EDGJLOAD" on page 217).
9. Run EDGJVERR job to create the CDS control record (see "Create DFSMSRmm CDS Control Record" on page 219) and run EDGUTIL to verify the DFSMSrmm CDS (see "Verifying the DFSMSrmm CDS" on page 220). If there are errors that do not allow you to start DFSMSrmm subsystem in RECORD mode, you can try to correct the errors and go back to step 7, change the necessary EDGCNVT control statements, and restart the conversion steps.
10. Start DFSMSrmm subsystem in RECORD mode.
11. Manually update any VRS definitions and save changes.
12. Execute DFSMSrmm housekeeping process (VRSEL, EXPROC and DSTORE) to assign storage locations and bin numbers.
13. Compare DFSMSrmm scratch and storage location reports with corresponding Control-M/Tape reports.
14. Correct any conversion problems and repeat steps 6 to 14.
15. Modify and install any required installation exits, such as EDGUX100 (the EDGCVRSX sample).
16. Start DFSMSrmm in WARNING mode.
17. When no unexplained differences, start DFSMSrmm in PROTECT mode and remove Control-M/Tape from the system.

The following table lists the conversion samples, programs, and JCL for the extraction of data from Control-M/Tape.

Table 51. DFSMSrmm supplied conversion samples, programs, and JCL

Name	Location	Function
EDGRCTT1	WEB	REXX source to create the extended Control-M/Tape record
EDGRCTT2	WEB	REXX source for Control-M/Tape media data base (MDB) extract

Table 51. DFSMSrmm supplied conversion samples, programs, and JCL (continued)

Name	Location	Function
EDGRCTT3	WEB	REXX source for Control-M/Tape Rules extract
EDGJCTT1	WEB	JCL to execute EDGRCTT1
EDGJCTT2	WEB	JCL to execute EDGRCTT2
EDGJCTT3	WEB	JCL to execute EDGRCTT3
EDGCNVT	LINKLIB	The EDGCNVT program
EDGCNXIT	SAMPLIB	Source for sample exit EDGCNVT
EDGCEBIN	LINKLIB	The EDGCEBIN program
EDGJCNTV	SAMPLIB	JCL to execute EDGCNVT and EDGCEBIN
EDGCVOVL	LINKLIB	The EDGCVOVL program
EDGJLOAD	SAMPLIB	JCL to execute EDGCVOVL, create and load CDS
EDGJVERR	SAMPLIB	JCL to create a CDS control record and execute EDGUTIL VERIFY processing

The sample JCL is set up to work in a "typical" installation. Some of the sample JCL is commented out so that you choose to use it or not as is your preference.

Data Extraction

The data extraction execs are designed to extract relevant information from the Control-M/Tape extract file. Then they convert the information about retention and vaulting contained in the RULES data set.

Control-M/Tape MDB records

Start the conversion by unloading the Control-M/Tape MDB records using the IOADUL utility shipped together with the product (this utility replaces the CTTDBDUL utility as of release 6.1.00). The figure below shows a sample JCL for using IOADUL utility which is used by the CTTDBDUL job in the Control-M/Tape JCL library.

```
//CTTDBDUL EXEC CTTDBDUL,DBFILEI=MDBD,DBFILEO=SEQ
//DAOUT DD DISP=(,CATLG),DSN=STSGWD.MBBD.DATA,
// UNIT=SYSDA,SPACE=(CYL,(10,10),RLSE),
// DCB=(LRECL=660,BLKSIZE=0,RECFM=FB)
```

Figure 41. Unload Control-M/Tape MDB using IOADUL utility

Note: In Control-M/Tape release 6.1.00 the record layout of the MDB has been modified and enlarged to 660 bytes. If you convert from one of the previous releases then specify in the JCL one of the following logical record length:

- LRECL = 460 for unloading Control-M/Tape release 5.14.xx MDB
- LRECL = 600 for unloading Control-M/Tape release 6.0.xx MDB

Create an extended Control-M/Tape record (EDGRCTT1)

The EDGRCTT1 exec creates an extended Control-M/Tape record including all needed information for the future conversion process. The figure below shows a sample JCL for EDGRCTT1 program.

```

//SORTVOL EXEC PGM=SORT,REGION=6M
//SYSPRINT DD  SYSOUT=*
//SORTIN  DD  DISP=SHR,DSN=STSGWD.MDBD.DATA
//SORTOUT DD  DISP=(,PASS),DSN=&&VOLUME,
//          UNIT=SYSDA,DCB=*.SORTIN,
//          SPACE=(TRK,(900,250),RLSE)
//SYSOUT  DD  SYSOUT=*
//SYSIN   DD  *

*   the following SORT statement is for Control-T 5.14.xx
    OPTION NOCHALT
    SORT FIELDS=(6,6,CH,A)

*   the following SORT statement is for Control-T 6.0.xx
*   SORT FIELDS=(7,6,CH,A)

*   the following SORT statement is for Control-T 6.1.xx
*   SORT FIELDS=(14,6,CH,A)

*   the following INCLUDE statement is for Control-T < 6.1.xx
    INCLUDE COND=(1,1,CH,EQ,C'V')

*   the following INCLUDE statement is for Control-T 6.1.xx
*   INCLUDE COND=(6,1,CH,EQ,C'V')

/*

```

Figure 42. Sample JCL to Execute EDGRCTT1 (Part 1 of 4)

```

//SORTDSN EXEC PGM=SORT,REGION=6M
//SYSPRINT DD  SYSOUT=*
//SORTIN  DD  DISP=SHR,DSN=STSGWD.MDBD.DATA
//SORTOUT DD  DISP=(,PASS),DSN=&&DSNAME,
//          UNIT=SYSDA,DCB=*.SORTIN,
//          SPACE=(TRK,(900,250),RLSE)
//SYSOUT  DD  SYSOUT=*
//SYSIN   DD  *

*   the following SORT statement is for Control-T 5.14.xx
    OPTION NOCHALT
    SORT FIELDS=(51,6,CH,A,49,2,CH,A)
*   the following SORT statement is for Control-T 6.0.xx
*   SORT FIELDS=(53,6,CH,A,51,2,CH,A)

*   the following SORT statement is for Control-T 6.1.xx
*   SORT FIELDS=(59,6,CH,A,57,2,CH,A)

*   the following INCLUDE statement is for Control-T < 6.1.xx
    INCLUDE COND=(1,1,CH,EQ,C'D')

*   the following INCLUDE statement is for Control-T 6.1.xx
*   INCLUDE COND=(6,1,CH,EQ,C'D')

/*

```

Figure 42. Sample JCL to Execute EDGRCTT1 (Part 2 of 4)

```

//EXTENDMD EXEC PGM=IKJEFT01,COND=(0,NE)
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DSN DD DISP=(OLD,DELETE),DSN=&&DSNAME
//VOL DD DISP=(OLD,DELETE),DSN=&&VOLUME
//EXTEXTR DD DISP=(,PASS),DSN=&&EXTENDED,
// UNIT=SYSALLDA,SPACE=(TRK,(900,250),RLSE),
// DCB=(LRECL=1200,BLKSIZE=6000,RECFM=FB)
//SYSTSIN DD *
EX 'SYS1.SEDGEXE1(EDGRCTT1)'
/*

```

Figure 42. Sample JCL to Execute EDGRCTT1 (Part 3 of 4)

```

//SORTEXTN EXEC PGM=SORT,REGION=6M
//SYSPRINT DD SYSOUT=*
//SORTIN DD DISP=(OLD,DELETE),DSN=&&EXTENDED
//SORTOUT DD DISP=(,CATLG),
// DSN=STSGWD.EDGRCTT1.EXTENDED.DATA,
// UNIT=SYSDA,DCB=*.SORTIN,
// SPACE=(TRK,(900,250),RLSE)
//SYSOUT DD SYSOUT=*
//SYSIN DD *

* the following SORT statement is for Control-T 5.14.xx
OPTION NOCHALT
SORT FIELDS=(115,6,CH,A,101,2,BI,A,649,2,BI,A)

* the following SORT statement is for Control-T 6.0.xx
* SORT FIELDS=(121,6,CH,A,107,2,BI,A,651,2,BI,A)

* the following SORT statement is for Control-T 6.1.xx
* SORT FIELDS=(122,6,CH,A,108,2,BI,A,652,2,BI,A)

/*

```

Figure 42. Sample JCL to Execute EDGRCTT1 (Part 4 of 4)

Input Data Sets

EDGRCTT1 has two input data sets.

Table 52. EDGRCTT1 Input Data Sets

DDNAME	Description
VOL	The unloaded Control-M/Tape MDB volume records (sorted).
DSN	The unloaded Control-M/Tape MDB data set records (sorted).

Output Data Sets

EDGRCTT1 creates only one output data sets. This data set is needed for the EDGRCTT2 extraction program.

Table 53. EDGRCTT1 Output Data Sets

DDNAME	Description
EXTEXTR	Contains all the created extended Control-M/Tape MDB records.

Return Codes

EDGRCTT1 sets a return code when it ends:

Table 54. EDGRCTT1 Return Codes

Return code	Description
0	Normal processing, the function was executed.
4	Normal completion. No errors detected One or more error messages issued. Valid records for input to EDGRCTT2 are produced, but the error messages should be reviewed to ensure that results are as expected.
12	Unsupported format of input data.
20	An error has been encountered in the exec processing. Refer to the REXX error messages for resolution.

Messages

CTT1000I EDGRCTT1 started *day month year hh mm ss*

Problem determination: where

day day the function started

month month the function started

year year the function started

hhmmss time the function started

Severity: Information

Explanation: None

System action: None

User response: None

CTT1001W No data set record found for volume *vvvvvv*

Problem determination: where

vvvvvv volume serial number

Severity: Warning

Explanation: No data set record is available for the processed volume.

System action: The zero volume sequence number is set to 1 to prevent possible later errors.

User response: Use the information from Control-M/Tape MDB to determine if the volume should have a data set, and correct it if required.

CTT1002W No volume record found for volume *vvvvvv* corresponding volume record created

Problem determination: where

vvvvvv volume serial number

Severity: Warning

Explanation: A data set record is found but no corresponding volume record was found.

System action: The corresponding volume record for this volume is created.

User response: Use the information from Control-M/Tape MDB to determine that created volume is as expected. Use DFSMSrmm CHANGEVOLUME command to correct it if required.

CTT1003E Unknown format of input data. Logical record length should be 460, 600 or 660 bytes

Severity: Error

Explanation: The input data set has an incorrect logical record length.

System action: Rerun the Control-M/Tape IOADUL program with the appropriate LRECL parameter for output data set (see "Control-M/Tape MDB records" on page 116).

User response: None

Extracting data from the extended MDB records (EDGRCTT2)

This section describes how to setup and control the Control-M/Tape MDB data extraction program. The figure below shows a sample JCL for EDGRCTT2 program.

```
//EXTRMDBD EXEC PGM=IKJEFT01,COND=(0,NE)
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//IN DD DISP=SHR,DSN=STSGWD.EDGRCTT1.EXTENDED.DATA
//CTTPARM DD DISP=SHR,DSN=STSGWD.PARM.DATA(CTTPARM)
//DEXTRACT DD DISP=(,CATLG),DSN=STSGWD.EDGRCTT2.DEXTRACT.DATA,
// UNIT=SYSALLDA,SPACE=(TRK,(900,250),RLSE),
// DCB=(RECFM=VB,LRECL=1024,BLKSIZE=0)
//DEXTOWN DD DISP=(,CATLG),DSN=STSGWD.EDGRCTT2.DEXTOWN.DATA,
// UNIT=SYSALLDA,SPACE=(TRK,(100,10),RLSE),
// DCB=(RECFM=VB,LRECL=1024,BLKSIZE=0)
//DEXTVRS DD DISP=(,CATLG),DSN=STSGWD.EDGRCTT2.DEXTVRS.DATA,
// UNIT=SYSALLDA,SPACE=(TRK,(900,250),RLSE),
// DCB=(RECFM=VB,LRECL=1024,BLKSIZE=0)
//CCARDS DD DISP=(,CATLG),DSN=STSGWD.EDGRCTT2.CCARDS.DATA,
// UNIT=SYSALLDA,SPACE=(TRK,(100,10),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//ERROR DD DISP=(,CATLG),DSN=STSGWD.EDGRCTT2.ERRORS.DATA,
// UNIT=SYSALLDA,SPACE=(TRK,(900,250),RLSE),
// DCB=(RECFM=FB,LRECL=133,BLKSIZE=0)
//REPORT DD DISP=(,CATLG),DSN=STSGWD.EDGRCTT2.REPORT.DATA,
// UNIT=SYSALLDA,SPACE=(TRK,(900,250),RLSE),
// DCB=(RECFM=FB,LRECL=133,BLKSIZE=0)
//CUSTOM DD *
/*USE UNITNAME FOR VMEDIA
/*CONVERT DELETED VOLUMES
IF MVDEF EQUALS ' ' THEN CATALOG
IF MVLOC EQUALS ' ' THEN HOME
IF VMEDIA EQUALS ' ' THEN CART3590
IF OWNER EQUALS ' ' THEN CONVERTS
CREATE DEFAULT VRS
EDM DSNAME PREFIX = VALUE1 VALUE2 VALUEN
//SYSTSIN DD *
EX 'SYS1.SEDGEXE1(EDGRCTT2)'
/*
```

Figure 43. Sample JCL to Execute EDGRCTT2

Input Data Sets

EDGRCTT2 has 2 input data sets:

Table 55. EDGRCTT2 Input Data Sets

DDNAME	Description
IN	Contains sorted extended Control-M/Tape MDB records.
SYSIN	Series of statements to tailor the processing.

Use the Control-M/Tape installation parameter member **CTTPARM** to handle the following CTTPARM values:

OVERJCL

EDGRCTT2 creates EDGCNVT control card 'VRS XDI ON/OFF' based on the Control-M/Tape OVERJCL parameter. For 'Y' VRS XDI ON is used.

CYCLECNT

EDGRCTT2 and EDGRCTT3 set the CYCLE/BYDAYSCYCLE retention type basing on the Control-M/Tape CYCLECNT parameter. Any value other than NONE causes CYCLEBYDAYS.

GRACECAT

EDGRCTT2 creates the additional NAME(GRACECAT) vital record specification to add to those VRS chains which end with WHILECATALOG retention. The reason is to ensure that all files with catalog retention are retained the specified number of additional days after they become uncataloged.

GRACECYC

EDGRCTT2 creates the additional NAME(GRACECYC) vital record specification to add to those VRS chains which end with cycle (CYCLE/BYDAYSCYCLE) retention. This is to ensure that all files with cycle retention are retained the specified number of additional days after they become not kept by other VRSes in the chain any more.

Additionally, you can manually handle some other CTTPARM parameters for conversion to DFSMSrmm parmlib options and functions:

DEFEXPDT

Use DFSMSrmm parmlib member OPTION RETPD to specify the same default retention in days as specified in your DEFEXPDT parameter.

CTLGWAIT

Use DFSMSrmm parmlib member OPTION CATRETPD(hours) as the equivalent to CTLGWAIT parameter. Multiply the CTLGWAIT value, which specifies the number of days to wait, by 24 to convert to hours because the value for CATRETPD is in hours.

VLTBYS1

DFSMSrmm has no direct equivalent to VLTBYS1 parameter. However, you can use LOCATION priority and VRS priority to establish your priorities for movement of the tape volumes. DFSMSrmm processes retention on each data set on the volume.

RTNUPD

DFSMSrmm VRSEL processing is similar to the update retention type ALL.

EXPDTDDN

Use this value to replace the 'NOTKEYD8' DD name which is hard-coded in the EDGUX100 sample exits (EDGUX100 and EDGCVR SX).

EXPDTYPE

If the expiration date type EXPDTYPE specifies CA1 or TLMS, refer to the samples EDGUX100 and EDGCVR SX which allow to support the special dates.

SLOTRNG1

DFSMSrmm uses the LOCDEF MEDINAME to handle different ranges of slots based on media shape.

Use the CUSTOM/SYSIN DD-statement to:

1. Set a default management value, if no other matched
2. Specify the default location for all management value VRS definitions
3. Set a default medianame
4. Set a default owner, if no valid owner is found

5. Specify creation of default DSNAME VRS ** with WHILECATALOG retention
6. Specify setting medianame basing on Control-M/Tape unit name value (by default one of the TAPE1600, TAPE6250, REDWOOD, CART3480, CART349E, CART3590 or CART3592 will be set depending on Control-M/Tape density of the data on the volume value).
7. Specify how to convert volumes marked as deleted in Control-M/Tape MDB. With this parameter they will be converted to DFSMSrmm scratch volumes. Otherwise deleted volumes will be ignored.
8. Specify the data set name prefixes separated by blanks to be used to determine volumes which have just one file and convert them with data set recording flag set to OFF.

Output Data Sets

EDGRCTT2 creates 6 different output data sets. These data sets are needed as input to the EDGCNVT conversion program:

Table 56. EDGRCTT2 Output Data Sets

DDNAME	Description
CCARDS	Contains all created EDGCNVT SYSIN file statements that can be used for EDGCNVT processing.
DEXTRACT	Contains all created library and data set records for input to the EDGCNVT conversion program.
DEXTOWN	Contains all created owner records for input to the EDGCNVT conversion program.
DEXTVRS	Contains all created vital record specification records for input to the EDGCNVT conversion program.
ERROR	Contains warning and error messages associated with the data extraction process.
REPORT	Contains informational messages associated with the data extraction process.

Return Codes

EDGRCTT2 sets a return code when it ends:

Table 57. EDGRCTT2 Return Codes

Return code	Description
0	Normal completion. No errors detected.
4	One or more warning messages issued. Valid records for input to EDGCNVT are produced, but the warning messages should be reviewed to ensure that results are as expected
8	One or more error messages issued. Unless you correct the reported error you will not get expected results under RMM. The errors should be corrected and the conversion process rerun.
12	Unsupported format of input data.
20	An error has been encountered in the exec processing. Refer to the REXX error messages for resolution.

Messages

CTT2000I EDGRCTT2 started *day month year hh mm ss*

Problem determination: where

day day the function started

month month the function started

year year the function started

hhmmss time the function started

Severity: Information

Explanation: None

System action: None

User response: None

CTT2001I Volser *pppppp* and *vvvvvv* are part of multi-volume sequence containing multiple data sets *dsn1 dsn2*

Problem determination: where

pppppp previous volume serial number

vvvvvv current volume serial number

dsn1 last data set on *pppppp*

dsn2 first data set on *vvvvvv*

Severity: Information

Explanation: The EDGRCTT2 conversion program has detect that the current processed volume serial is part of a multi-volume sequence containing multiple data sets.

System action: None

User response: None

CTT2002W Owner *00000000* has more than 8 chars

Problem determination: where

00000000
current volume owner

Severity: Warning

Explanation: The current processed owner information has more than 8 characters.

System action: The default owner is used.

User response: Check the Control-M/Tape MDB, correct the error and rerun the conversion process if required.

CTT2003W Owner *00000000* has one or more invalid chars

Problem determination: where

00000000
current volume owner

Severity: Warning

Explanation: The current processed owner information has one or more invalid character(s). An owner must follow the MVS standard naming convention.

System action: The default owner is used.

User response: Check the Control-M/Tape MDB, correct the error and rerun the conversion process if required.

CTT2004W Owner *00000000* has an invalid first character

Problem determination: where

00000000
current volume owner

Severity: Warning

Explanation: The current processed owner information is not starting with a character or national character.

System action: The default owner is used.

User response: Check the Control-M/Tape MDB, correct the error and rerun the conversion process if required.

CTT2005W Owner 00000000 has more than 8 chars

Problem determination: where

00000000

current volume owner

Severity: Warning

Explanation: The current processed owner information has more than 8 characters. The HLQ is used instead of the owner filed, because the owner field was empty.

System action: The default owner is used.

User response: Check the Control-M/Tape MDB, correct the error and rerun the conversion process if required.

CTT2006W Owner 00000000 has one or more invalid chars

Problem determination: where

00000000

current volume owner

Severity: Warning

Explanation: The current processed owner information has one or more invalid character(s). An owner must following the MVS standard naming convention. The HLQ is used instead of the owner filed, because the owner field was empty.

System action: The default owner is used.

User response: Check the Control-M/Tape MDB, correct the error and rerun the conversion process if required.

CTT2007W Owner 00000000 has an invalid first character

Problem determination: where

00000000

current volume owner

Severity: Warning

Explanation: The current processed owner information is not starting with a alpha or national character. The HLQ is used instead of the owner filed, because the owner field was empty.

System action: The default owner is used.

User response: Check the Control-M/Tape MDB, correct the error and rerun the conversion process if required.

CTT2008W Different expdt found on volume vvvvvv Type ddsexpt1 CTTDVL=dvlexpd, CTTDDS=ddsexpd1 now used

Problem determination: where

vvvvvv current volume serial number

ddsexpt1

current volume expiration type 1

dvlexpd current volume expiration date

ddsexpd1

current data set expiration

Severity: Warning

Explanation: In the current processed extended MDB record different volume and data set retention was detected.

System action: The data set expiration date is used.

User response: Check the Control-M/Tape MDB, correct the error and rerun the conversion process if required.

CTT2009W Different expdt found on volume *vvvvvv* Type *ddsexpt2* CTTDVL=*dvolexpd*, CTTDDS=*ddsexpd1* now used

Problem determination: where

vvvvvv current volume serial number

ddsexpt2

current volume expiration type 2

dvolexpd current volume expiration date

ddsexpd1

current data set expiration

Severity: Warning

Explanation: In the current processed extended MDB record different volume and data set retention was detected.

System action: The data set expiration date is used.

User response: Check the Control-M/Tape MDB, correct the error and rerun the conversion process if required.

CTT2010E Invalid slot number *ssssss* for volume *vvvvvv* found. Slot changed to 000000

Problem determination: where

ssssss current slot number

vvvvvv current volume serial number

Severity: Error

Explanation: In the current processed extended MDB record invalid slot number was detected.

System action: The 000000 slot number is used.

User response: Check the Control-M/Tape MDB, correct the error and rerun the conversion.

CTT2011E Invalid special expdt *dxsjexpd* found on volume *vvvvvv*

Problem determination: where

vvvvvv current volume serial number

dxsjexpd

special expiration date

Severity: Error

Explanation: In the current processed extended MDB record has a invalid special expiration date.

System action: No VRS record is created for this volume.

User response: Check the Control-M/Tape MDB, correct the error and rerun the conversion process.

CTT2012E the next volume *nnnnnn* pointed by volume *vvvvvv* does not exist

Problem determination: where

nnnnnn serial number of next volume of multi-volume chain

vvvvvv serial number of current processed volume of multi-volume chain

Severity: Error

Explanation: The missing next volume *nnnnnn* pointed by volume *vvvvvv* was detected.

System action: The volume next to nnnnnn will be processed if exist. The sequence will be different from expected.

User response: Check the Control-M/Tape MDB, correct the error and rerun the conversion process.

CTT2013I location information for library *vloc* created

Problem determination: where

vloc current location

Severity: Information

Explanation: Control-M/Tape volume or VRS record contains location information which should be defined to DFSMSrmm.

System action: EDGCNVT LOCDEF control statement created.

User response: None

CTT2014I VRS record for data set *dsname* created

Problem determination: where

dsname current data set name

Severity: Information

Explanation: Control-M/Tape data set name record contains specific date (90000-90365 and 98000-99366).

System action: Corresponding VRS management class created.

User response: None

CTT2015I owner record for owner *000000* created

Problem determination: where

000000 current owner

Severity: Information

Explanation: Control-M/Tape volume or data set record contains owner information which should be defined to DFSMSrmm.

System action: Owner record created.

User response: None

CTT2016I You have used a *ddname* DD statement

Problem determination: where

ddname Name of DD-statement to customize a conversion process (one of CUSTOM or SYSIN)

Severity: Information

Explanation: EDGJCTT2 has CUSTOM/SYSIN DD-statement for tailoring a conversion process.

System action: Specified CUSTOM/SYSIN DD-statement will be processed.

User response: None

CTT2017I You have specified to set the default VRS management value to *defmv*

Problem determination: where

defmv default VRS management value

Severity: Information

Explanation: Default MV specified in CUSTOM/SYSIN DD-statement will be used if no other matched.

System action: Default MV set.

User response: None

CTT2018I You have specified to set the default location for all management values to *locmv*

Problem determination: where

locmv default location for all MV VRS definitions

Severity: Information

Explanation: Default location specified in CUSTOM/SYSIN DD-statement will be used for all vital record specification management value definitions if no other matched.

System action: Default location for VRS management value definitions set.

User response: None

CTT2019E *ddname* control card not processed: *ccard*

Problem determination: where

ddname Name of DD-statement to customize a conversion process (one of CUSTOM or SYSIN)

ccard control card contents

Severity: Error

Explanation: The current processed control card contains invalid contents.

System action: Control card is not processed.

User response: Check the EDJCTT2 CUSTOM/SYSIN DD-statements, correct the error and rerun the job.

CTT2020E Incorrect format of extended extract file

Severity: Error

Explanation: The extended extract file has incorrect control record. This record needs to determine Control-M/Tape release.

System action: Rerun the EDGRCTT1 program.

User response: None

CTT2021I You have specified to set the default MEDIANAME for all volumes to *medianame*

Problem determination: where

medianame

default MEDIANAME value

Severity: Information

Explanation: Default MEDIANAME specified in CUSTOM/SYSIN DD-statement will be used if no other matched.

System action: Default MEDIANAME set.

User response: None

CTT2022I You have specified to set the default owner ID to *ownername* if no valid owner is found

Problem determination: where

ownername

default OWNERNAME value

Severity: Information

Explanation: Default OWNERNAME specified in CUSTOM/SYSIN DD-statement will be used if no other matched.

System action: Default OWNERNAME set.

User response: None

CTT2023I You have specified to create the default DSNAME VRS ** with WHILECATALOG retention

Severity: Information

Explanation: Creation of default VRS DSNAME('**') is requested.

System action: Default VRS DSNAME('**') will be created.

User response: None

CTT2024W Volume sequence for volume vvvvvv changed from 0000 to 0001

Problem determination: where

vvvvvv current volume serial number

Severity: Warning

Explanation: The EDGRCTT2 conversion program has detect that the current processed volume serial has a incorrect volume sequence 0. The volume sequence have been changed to 1.

System action: None

User response: None

CTT2025I You have specified to set the MEDIANAME basing on Control-T unit name

Severity: Information

Explanation: The DFSMSrmm volume media name will be set basing on Control-M/Tape unit name value. By default the EDGRCTT2 conversion program sets volume media name basing on Control-M/Tape density.

System action: None

User response: None

CTT2026I You have specified to convert volumes marked as deleted in Control-T to RMM scratch volumes

Severity: Information

Explanation: The volumes marked as deleted in Control-T MDB will be converted to DFSMSrmm scratch volumes. Without specifying this parameter they will be ignored.

System action: None

User response: None

CTT2027I You have used a CTTPARM DD statement

Severity: Information

Explanation: EDGJCTT2 has CTTPARM DD which specifies Control-M/Tape installation parameter member.

System action: Control-M/Tape installation parameter member will be processed.

User response: None

CTT2029I CTTPARM parameter processed: ppppppppp

Problem determination: where

ppppppppp
processing string

Severity: Information

Explanation: The processing Control-M/Tape installation parameter is supported by conversion program.

System action: None

User response: None

CTT2030I EDGRCTT2 will treat all CYCLES as BYDAYSCYCLE

Severity: Information

Explanation: The retention type will be set to BYDAYSCYCLE.

System action: None

User response: None

CTT2031I CTTPARM parameter not processed: *pppppppp*

Problem determination: where

pppppppp
processing string

Severity: Information

Explanation: The processing Control-M/Tape installation parameter is not supported by conversion program.

System action: None

User response: None

CTT2032I volume *vvvvvv* not converted because it marked as deleted in Control-T

Problem determination: where

vvvvvv current volume serial number

Severity: Information

Explanation: The processing volume is marked as deleted in the Control-M/Tape MDB.

System action: The volume is not converted. You can specify to convert deleted volumes (see 7 on page 122).

User response: None

CTT2033W The specified data set name prefix *prefix* has more than 44 allowable characters

Problem determination: where

prefix data set name prefix

Severity: Warning

Explanation: The too long data set name prefix is used for an 'EDM DSNAME PREFIX =' statement in CUSTOM DD.

System action: The incorrectly specified data set name prefix is ignored.

User response: Check the CUSTOM DD-statements, correct the error and rerun the conversion process if required.

CTT2034I You have specified DSNAME prefix *prefix* to be used for converting EDM volumes

Problem determination: where

prefix data set name prefix

Severity: Information

Explanation: DSNAME prefix specified in CUSTOM DD will be used for converting EDM volumes.

System action: None

User response: None

Extracting data from the RULES file (EDGRCTT3)

This section describes how to setup and control the Control-M/Tape Rules file extraction program. The figure below shows a sample JCL for EDGRCTT3 program.

```
//EXTRRULE EXEC PGM=IKJEFT01,COND=(0,NE)
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//IN DD DISP=SHR,DSN=STSGWD.RULES.DATA(RETRULE)
//CTTPARM DD DISP=SHR,DSN=STSGWD.PARM.DATA(CTTPARM)
//DEXTRACT DD DISP=(,PASS),DSN=&&DEXTRACT,
// UNIT=SYSALLDA,SPACE=(TRK,(900,250),RLSE),
// DCB=(RECFM=VB,LRECL=1024,BLKSIZE=0)
//ERROR DD DISP=(,CATLG),DSN=STSGWD.EDGRCTT3.ERRORS.DATA,
// UNIT=SYSALLDA,SPACE=(TRK,(900,250),RLSE),
// DCB=(RECFM=FB,LRECL=133,BLKSIZE=0)
//REPORT DD DISP=(,CATLG),DSN=STSGWD.EDGRCTT3.REPORT.DATA,
// UNIT=SYSALLDA,SPACE=(TRK,(900,250),RLSE),
// DCB=(RECFM=FB,LRECL=133,BLKSIZE=0)
/* ----- *
/* You can use the SYSIN DD statement to: *
/* 1. specify a default OWNER for volumes w/o valid owner *
/* ----- *
//CUSTOM DD * IF OWNER EQUALS ' ' THEN CONVERS
//SYSIN DD * EX 'SYS1.SEDGEXE1(EDGRCTT3)'
/*
//SORT EXEC PGM=SORT,REGION=6M
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=(OLD,DELETE),DSN=&&DEXTRACT
//SORTOUT DD DISP=(,CATLG),
// DSN=STSGWD.EDGRCTT3.DEXTRACT.DATA,
// UNIT=SYSDA,DCB=*.SORTIN,
// SPACE=(TRK,(900,250),RLSE)
//SYSIN DD *
OPTION NOCHALT
SORT FIELDS=(1,80,CH,A),DYNALLOC
SUM FIELDS=NONE removes duplicates
END
/*
```

Figure 44. Sample JCL to Execute EDGRCTT3

Input Data Sets

EDGRCTT3 has one input data set:

Table 58. EDGRCTT3 Input Data Sets

DDNAME	Description
IN	Contains Control-M/Tape rules definitions.
SYSIN	Series of statements to tailor the processing.

Use the CUSTOM DD-statement to:

- Set a default owner, if no valid owner is found

Use the Control-M/Tape installation parameter member **CTTPARM** to handle the following CTTPARM values:

CYCLECNT

EDGRCTT2 and EDGRCTT3 set the CYCLE/BYDAYSCYCLE retention type basing on the Control-M/Tape CYCLECNT parameter. Any value other than NONE causes CYCLEBYDAYS.

DEFABEND

EDGRCTT3 creates the special ABEND retention vital record specification basing on this parameter.

GRACECAT

EDGRCTT3 initializes the next VRS field of the last VRS in a VRS chain which ends with WHILECATALOG retention. The reason is to ensure that all files with catalog retention are retained the specified number of additional days after they become uncataloged. The corresponding NAME(GRACECAT) vital record specification is created by EDGRCTT2 program.

GRACECYC

EDGRCTT3 initializes the next VRS field of the last VRS in a VRS chain which ends with cycle (CYCLE/BYDAYSCYCLE) retention. This is to ensure that all files with cycle retention are retained the specified number of additional days after they become not kept by other VRSeS in the chain any more.

Additionally, you can manually handle some other CTTPARM parameters for conversion to DFSMSrmm parmlib options and functions. For more information refer back to the same chapter for EDGRCTT2 program Figure 43 on page 120.

Output Data Sets

EDGRCTT3 creates 3 different output data sets.

Table 59. EDGRCTT3 Output Data Sets

DDNAME	Description
DEXTRACT	Contains all created owner and vital records for input to the EDGCNVT conversion program.
ERROR	Contains warning and error messages associated with the data extraction process.
REPORT	Contains informational messages associated with the data extraction process.

Return Codes

EDGRCTT3 sets a return code when it ends:

Table 60. EDGRCTT3 Return Codes

Return code	Description
0	Normal completion. No errors detected.
4	One or more warning messages issued. Valid records for input to EDGCNVT are produced, but the warning messages should be reviewed to ensure that results are as expected.
8	One or more error messages issued. Unless you correct the reported error you will not get expected results under RMM. The errors should be corrected and the conversion process rerun.
20	An error has been encountered in the exec processing. Refer to the REXX error messages for resolution.

Messages

CTT3001E Record not processed *record*

Problem determination: where

record current rule record

Severity: Error

Explanation: The EDGRCTT3 conversion program has detect a undefined record type or a record type W.

System action: The current record is not processed.

User response: Check the Control-M/Tape rules file, correct the error and rerun the conversion process.

CTT3002I old DSN mask 00000 changed new DSN mask nnnnn

Problem determination: where

00000 Control-M/Tape data set name mask

nnnnn DFSMSrmm data set name mask

Severity: Information

Explanation: The current processed data set name mask has 8 characters in the last level qualifier and an asterisk at the end character.

System action: The data set name mask is changed to matching DFSMSrmm data set name mask conventions. In this case a '.*' replaces at the end.

User response: None

CTT3003E DSN level lllll has more than 8 characters

Problem determination: where

lllll current checked data set level

Severity: Error

Explanation: One of the data set name levels has more than 8 characters.

System action: The current record is not processed.

User response: Check the Control-M/Tape MDB, correct the error and rerun the conversion process.

CTT3004E DSN level lllll has an invalid character

Problem determination: where

lllll current checked data set level

Severity: Error

Explanation: One of the data set name levels has an invalid character.

System action: The current record is not processed.

User response: Check the Control-M/Tape MDB, correct the error and rerun the conversion process.

CTT3005E DSN level lllll has an invalid first character

Problem determination: where

lllll current checked data set level

Severity: Error

Explanation: The current processed data set level is not starting with a alpha or national character.

System action: The current record is not processed.

User response: Check the Control-M/Tape MDB, correct the error and rerun the conversion process.

CTT3006I old DSN mask 00000 changed new DSN mask nnnnn

Problem determination: where

00000 Control-M/Tape data set name mask

nnnnn DFSMSRmm data set name mask

Severity: Information

Explanation: The current processed data set name mask has no asterisk at the end.

System action: The data set name mask is changed to matching DFSMSRmm data set name mask conventions. In this case a '.*' is added at the end.

User response: None

CTT3007I old DSN mask 00000 changed new DSN mask nnnnn

Problem determination: where

00000 Control-M/Tape data set name mask

nnnnn DFSMSRmm data set name mask

Severity: Information

Explanation: The current processed data set name mask has no asterisk at the end.

System action: The data set name mask is changed to matching DFSMSRmm data set name mask conventions. In this case a '.G^^^V^^' is added as last qualifier.

User response: None

CTT3008I old DSN mask 00000 changed new DSN mask nnnnn

Problem determination: where

00000 Control-M/Tape data set name mask

nnnnn DFSMSRmm data set name mask

Severity: Information

Explanation: The current processed data set name mask has no asterisk at the end.

System action: The data set name mask is changed to matching DFSMSRmm data set name mask conventions. In this case a '.*' is added as the end.

User response: None

CTT3009W The count for rule changed to a value of 99999

Problem determination: where

rule current rule record

Severity: Warning

Explanation: The current processed rule record has invalid count value.

System action: The count value is changed to a value 99999.

User response: Check the Control-M/Tape MDB, correct the error and rerun the conversion process if required.

CTT3010I owner record for owner 00000000 created

Problem determination: where

00000000

Control-M/Tape current rule record owner

Severity: Information

Explanation: Owner information for processed rule record has been created.

System action: None

User response: None

CTT3011I You have used a CUSTOM DD statement

Severity: Information

Explanation: EDGJCTT3 has CUSTOM DD-statement for tailoring a conversion process.

System action: Specified CUSTOM DD-statement will be processed.

User response: None

CTT3012I You have specified to set the default owner ID to 000000 if no valid owner is found

Problem determination: where

000000 default owner

Severity: Information

Explanation: Default owner specified in CUSTOM DD-statement will be used if no valid owner is found.

System action: None

User response: None

CTT3013E CUSTOM control card not processed: ccard

Problem determination: where

ccard control card contents

Severity: Error

Explanation: The current processed control card contains invalid contents.

System action: Control card is not processed.

User response: Check the EDGJCTT3 CUSTOM DD-statements, correct the error and rerun the job.

CTT3014I Calendar facility for rule : rule not converted

Problem determination: where

rule current rule

Severity: Information

Explanation: Control-M/Tape calendar function has no DFSMSrmm equivalent.

System action: Control card is skipped.

User response: None

CTT3015I PROGRAM parameter for rule : rule not converted

Problem determination: where

rule current rule

Severity: Information

Explanation: Control-M/Tape PROGRAM rule parameter has no DFSMSrmm equivalent.

System action: Control card is skipped.

User response: None

CTT3016I You have used a CTTPARM DD statement

Severity: Information

Explanation: EDGJCTT3 has CTTPARM DD which specifies Control-M/Tape installation parameter member.

System action: Specified Control-M/Tape. installation parameter member will be processed.

User response: None

CTT3017I CTTPARM parameter processed: *pppppppp*

Problem determination: where

pppppppp
processing string

Severity: Information

Explanation: The processing Control-M/Tape installation parameter is supported by conversion program.

System action: None

User response: None

CTT3018I EDGRCTT3 will treat all CYCLES as BYDAYSCYCLE

Severity: Information

Explanation: The retention type for cycle data sets will be set to BYDAYSCYCLE.

System action: None

User response: None

CTT3019I CTTPARM parameter not processed: *pppppppp*

Problem determination: where

pppppppp
processing string

Severity: Information

Explanation: The processing Control-M/Tape installation parameter is not supported by conversion program.

System action: None

User response: None

Conversion Considerations

Volume Ownership

DFSMSrmm requires that all private volumes have an owner. DFSMSrmm uses the RACF user ID of the batch job creating the first file. In Control-M/Tape you have owner information in the volume and data set records and we will use this Control-M/Tape ownership concept, if no owner information is available we will use the default owner name as the volume owner. During conversion you can change the REXX code or use the EDGCNVT SYSIN file statements to change this concept. As tapes are recycled and reused, DFSMSrmm will record the new owner.

Tape Pool Management

Control-M/Tape allows you to define ranges of tape volumes in a tape library to a specific pool. It restricts scratch requests to a specific pool based on the creating user and/or data set name.

Control-M/Tape supports multiple scratch pool addressing by using a table. If the volume opened for output does not fall within the specified range, Control-M/Tape denies the request and dismounts the volume. If you are using this function, you must use the DFSMSrmm multiple scratch pools facility. With this support, you will be able to direct new tape data sets to specific scratch pools on the basis of job name and data set name. Thus you can discretely manage sets of volumes.

DFSMSrmm decides which pool to use on the basis of the VLPOOL definitions in the DFSMSrmm PARMLIB member. This function is dependent on the correct VLPOOL definitions, but the customer can decide which pool to select by using the EDGUX100 / EDGCVR SX installation-wide exit or SMS ACS routines.

You can use ACS routines to automatically determine the target storage group and assign data classes, storage classes, and management classes to SMS-managed data sets. You can use the pre-ACS interface to provide additional information like vault destination or pool information to the ACS routines. You can use the EDGUX100 installation exit for pre-ACS processing. When called during pre-ACS processing, values selected by the exit are used as input to the ACS routine. Then when the EDGUX100 installation exit is called at OPEN or when a volume is mounted, the exit can provide the same values it passed during the pre-ACS processing or provide different vital record specification management values and pool values. See "**z/OS DFSMSrmm Implementation and Customization Guide**" (SC26-7405) for more details.

DFSMSrmm intercepts the mount messages issued on the system and, for those nonspecific volume requests, calls the installation exit to enable it to make a decision. DFSMSrmm saves the scratch pool selected by the exit for use in updating tape drive displays, updating the mount message, and for validating the volume when one is mounted.

The exit can optionally request that DFSMSrmm prevent the cartridge loader from being indexed. Thus drives can be selectively preloaded with scratch volumes from specific pools, but scratch volumes from other pools can be manually mounted without the cartridge loader being emptied.

When the exit selects a pool, the only acceptable volumes are those in that pool. When DFSMSrmm selects the scratch pool without use of the exit, any scratch volume from any eligible scratch pool on that system is acceptable.

DFSMSrmm provides a working sample exit that implements a scratch pooling system based on job names and data set names; all you have to do is update a table containing job names, data set names, and pool prefixes in the EDGUX100 exit.

You can also specify different data set or job name pool tables for each system. If you specify the NAME operand in the VLPOOL command in PARMLIB member EDGRMMxx, the tape drive will display this name rather than the pool prefix.

Storage Location Management

You define and use your installation management policies for those tape volumes and data sets that contain installation-critical data for disaster recovery.

Because DFSMSrmm and Control-M/Tape differ in terms of how they define and process storage location management policies, it is important that you review the current storage location management policy to understand retention and movement criteria.

Control-M/Tape manages tape volume movement to and among vaults using your installation-defined vaulting policy. You store this policy in a table, which a program processes during the batch update, comparing the entry to the MDB volume record. When there is a match, the tape volume is marked for

movement to and among the vaults. The policy is based on data set name, fully or partial qualified, and further qualified by job name. Control-M/Tape supports many vaults and does not restrict movement to and among vaults.

The batch program automatically updates the MDB volume record for tape volumes meeting the criteria with the proper location ID (vault code) and generates the store bin (slot) numbers to use in the storage locations. In addition it generates a report that indicates the current location, the destination location, and the store bin numbers for the tape volumes that require movement. If you delete a retention or vaulting policy, Control-M/Tape continues to retain data based on the deleted entry.

Control-M/Tape uses the predefined outcode location name of LIBR or LIB to delay volume movement until the specified number of days is reached. You can accomplish the same with DFSMSrmm by defining a VRS with LOCATION(HOME) and use NEXTVRS to chain VRS definitions together. Do not use the DELAY option of VRS unless the delay is only to apply to the latest copy or cycle.

Using your installation-defined VRSs, DFSMSrmm automatically controls the movement of the data sets on a tape volume or the tape volume from the tape library to and among the storage locations. You can specify a data set name or a data set name mask and, optionally, a job name or job name mask to define a VRS. Additionally, you can specify a VOLSER or a generic VOLSER.

DFSMSrmm does not restrict the movement of the data set on the tape volume or the tape volume to and among the storage locations. When a data set on the tape volume, or the tape volume, is moved from a tape library to and among the storage locations for disaster recovery or vital records, DFSMSrmm assigns a storage location bin number, storage location name, and store date.

DFSMSrmm allows you to specify one location name in each data set or volume VRS. If you require that a data set or volume be moved to more storage locations, you can chain together two or more VRSs by using the NEXTVRS, NAME, and ANDVRS keywords. DFSMSrmm does not restrict any sequence of data set or volume moves among storage locations and system-managed libraries.

During inventory management processing, DFSMSrmm identifies the volumes that must be moved to and among the storage locations and assigns the exact store bin numbers where the volumes are to be stored. DFSMSrmm uses the retention criteria from the VRS to manage tape volume retention in the storage locations.

When you convert to DFSMSrmm, you can continue to use the same vault names, but, because DFSMSrmm supports longer names, you may want to change the vault names to be more meaningful.

Retention Management

Control-M/Tape uses specific or unique Julian dates (90000-90365 and 98000-99366) to control tape data set retention and expiration. It also provides the ability to differentiate the dates with true expiration dates using a unique JCL DD-statement when creating the tape data set.

In addition, Control-M/Tape allows you to change the tape data set's retention and expiration value, using the user-defined rules. With the user-defined rules data set, you can qualify by data set name, partial data set name, further qualify by job name, program name account information, owner, media name or volume serial number.

Control-M/Tape does not recognize that EXPDT dates of 90000-90365 and 98000-99364 have special meanings; but, by using the installation-wide exit, EDGUX100, or a management class with VRS, you can get DFSMSrmm to provide the same function. DFSMSrmm uses VRSs, which define the policy to control the tape data set retention and tape volume retention. Expiration dates 99365 and 99366 or 1999/365 and 1999/366 are considered 'never-scratch' dates in DFSMSrmm.

See the table below for management class VRSs created by EDGRCTT2 for Control-M/Tape expiration dates :

Table 61. Control-M/Tape retention methods and DFSMSrmm VRS equivalences

Control-M/Tape expiration date	DFSMSrmm VRS management value	Description
0090/000	CATALOG	while catalog retention
0090/001 - 0090/365	CATLGddd	number of days retention
0098/001 - 0098/365	LDATEDddd	days since last referenced
0099/000	CATALOG	while cataloged retention
0099/001 - 0099/365	CYCLEddd	number of cycles/copies retention
0099/999	PERM	permanent retention
9990/000	CATALOG	while catalog retention
9990/001 - 9990/365	CATLGccc	number of days retention
9998/001 - 9998/365	LDATEDddd	days since last referenced
9999/000	CATALOG	while cataloged retention
9999/001 - 0099/365	CYCLEccc	number of cycles/copies retention
9999/999	PERM	permanent retention

DFSMSrmm provides support for the specific and unique Julian dates of 98000-99364 having special meanings. The support is provided through an installation-wide exit, which you can use to interrogate and recognize the dates. When the installation-wide exit detects a data set using the Julian dates, it assigns a VRS management value that identifies the retention management policy. A sample EDGUX100, called EDGCVRSX, is available for download, that uses a dynamic table driven assignment method based on data set names and keyword dates.

Control-M/Tape treats DFSMSHsm as an EDM, so it does not maintain volume sequence information in its database for multi-volume DFSMSHsm dump tapes , where as it does for non-HSM-managed tapes. During the conversion process the extract program has no choice but to assume that each and every DFSMSHsm dump tape contains a single volume file instead of a multi-volume file. DFSMSrmm maintains volume sequence information for multi-volume DFSMSHsm dump tapes so it can recognize a set of dump volumes as being one cycle.

Any volume flagged with a EDM userid is converted with data set recording turned OFF.

Parallel Processing

The timing of the DFSMSrmm update to its CDS during OPEN processing is such that any volumes rejected by the vendor system should not result in updates to the DFSMSrmm CDS.

Also, the timing of the DFSMSrmm reject of a tape is such that recording by Control-M/Tape may not occur if DFSMSrmm rejects a volume. Therefore, parallel running problems are unlikely to be the result of volume rejection and control by one system or the other. More likely, the differences in inventory management functions such as expiration and retention control cause the problems. DFSMSrmm's uncatalog of tape data sets could cause the other vendor system to return a tape to scratch and vice versa.

Timing of inventory management and batch runs may vary, causing volumes to return to scratch at different times, or to move between storage locations at different times. This can cause problems if both systems are in protect mode at the same time. If DFSMSrmm is in record-only or warning mode, the differences should not be a significant problem because DFSMSrmm continues to allow tapes to be used and keeps track of the latest tape information.

Another method to consider for conversion is to run both DFSMSrmm and the other vendor product in protect mode in parallel, but use different ranges of volumes managed by each system. You can use the exits available in each product (EDGUX100 for DFSMSrmm) to control which volumes they manage and which volumes they are to ignore or treat as foreign. In time, as volumes return to scratch, or as new applications are added, you can move more and more volumes to DFSMSrmm control.

When running in parallel, you should prevent DFSMSrmm from uncatalog tape data sets, but continue to run both sets of inventory management. This can be accomplished by setting the PARMLIB option to UNCATALOG(N) in member EDGRMMxx.

Chapter 8. Conversion from ASG-ZARA/AUTOMEDIA

This chapter contains information about converting from ASG-ZARA/AutoMedia to DFSMSrmm. The conversion programs support AutoMedia release 1.3, and ASG-ZARA releases 1.4 and 1.5. For general conversion process questions refer to the Redbook "**Converting to DFSMSrmm from ASG-ZARA or AutoMedia**" (SG24-6275).

If you do not have the Redbook, we recommend you to order a copy. You can check for the Redbooks on: <http://www.redbooks.ibm.com> and search for 'RMM'.

The following steps summarize the recommended conversion process:

1. Review the "**Converting to DFSMSrmm from ASG-ZARA or AutoMedia**" (SG24-6275) , the rest of this chapter, and Chapter 9, "Building the DFSMSrmm CDS," on page 185.
2. Extract data from the ASG-ZARA/AutoMedia database, creating an extract file.
3. Familiarize yourself with the sample conversion jobs.
4. Define and customize the DFSMSrmm system.
5. Start DFSMSrmm subsystem in MANUAL mode to get familiarization with dialog and commands.
6. Run EDGJCZA1, EDGJCZA2 and EDGJCZA3 ASG-ZARA/AutoMedia conversion jobs to produce a valid input to the EDGCNVT program and check the messages produced.
7. Customize created EDGCNVT SYSIN control statements (see "SYSIN Control Statements" on page 191) and run EDGJCNVT job (see "Running EDGCNVT and EDGCEBIN" on page 187).
8. Run EDGJLOAD job to chain owner records to volume records and load the DFSMSrmm CDS (see "Run EDGJLOAD" on page 217).
9. Run EDGJVERR job to create the CDS control record (see "Create DFSMSRmm CDS Control Record" on page 219) and run EDGUTIL to verify the DFSMSrmm CDS (see "Verifying the DFSMSrmm CDS" on page 220).
If there are errors that do not allow you to start DFSMSrmm subsystem in RECORD mode, you can try to correct the errors and go back to step 7, change the necessary EDGCNVT control statements, and restart the conversion steps.
10. Start DFSMSrmm subsystem in RECORD mode.
11. Manually update any VRS definitions and save changes.
12. Execute DFSMSrmm housekeeping process (VRSEL, EXPROC and DSTORE) to assign storage locations and bin numbers.
13. Compare DFSMSrmm scratch and storage location reports with corresponding ASG-ZARA/AutoMedia reports.
14. Correct any conversion problems and repeat steps 6 to 14.
15. Modify and install any required installation exits, such as EDGUX100 (the EDGCVR SX sample).
16. Start DFSMSrmm in WARNING mode.
17. When no unexplained differences, start DFSMSrmm in PROTECT mode and remove ASG-ZARA/AutoMedia from the system.

The table below lists the samples for the extraction of data from ASG-ZARA/AutoMedia:

Table 62. DFSMSrmm-supplied samples for conversion from ASG-ZARA/AutoMedia

Name	Location	Function
HLASM modules		
EDGCNVT	LINKLIB	The EDGCNVT program

Table 62. DFSMSrmm-supplied samples for conversion from ASG-ZARA/AutoMedia (continued)

Name	Location	Function
EDGCEBIN	LINKLIB	The EDGCEBIN program
EDGCVOVL	LINKLIB	The EDGCVOVL program
EDGCNEXIT	SAMPLIB	Source for sample exit for EDGCNVT
REXX execs		
EDGRCZA1	WEB	Source for ASG-ZARA/AutoMedia conversion exec 1
EDGRCZA2	WEB	Source for ASG-ZARA/AutoMedia conversion exec 2
EDGRCZA3	WEB	Source for ASG-ZARA/AutoMedia conversion exec 3
JCLs		
EDGJCNT	SAMPLIB	JCL to execute EDGCNVT and EDGCEBIN
EDGJZA1	WEB	JCL to execute EDGRCZA1 ASG-ZARA/AutoMedia conversion exec
EDGJZA2	WEB	JCL to execute EDGRCZA2 ASG-ZARA/AutoMedia conversion exec
EDGJZA3	WEB	JCL to execute EDGRCZA3 ASG-ZARA/AutoMedia conversion exec
EDGJLOAD	SAMPLIB	JCL to execute EDGCVOVL, create and load CDS
EDGJVERR	SAMPLIB	JCL to create a CDS control record and execute EDGUTIL VERIFY processing

The sample JCL is set up to work in a "typical" installation. Some of the sample JCL is commented out so that you choose to use it or not as is your preference.

The data extraction execs are designed to extract relevant information from the ASG-ZARA/AutoMedia extract file. Then they convert the information about retention and vaulting contained in the RULES data set.

Unload the ASG-ZARA/AUTOMEDIA DB Records (ZARAXTRT)

Start the conversion by unloading the ASG-ZARA/AutoMedia DB records using the ZARAXTRT utility shipped together with the product.

The next figure shows a sample JCL for using ZARAXTRT utility:

```
//STEP1 EXEC ZARAXTRT,PARM='ZARA'
//SYSPRINT DD SYSOUT=*
//ZARAOUT DD DISP=(,CATLG),DSN=STSGWD.ZARA.DATA,
// UNIT=SYSDA,SPACE=(CYL,(30,10),RLSE),
// DCB=(LRECL=370,BLKSIZE=0,RECFM=VB)
//SYSIN DD *
DSN=ALL $$
```

Figure 45. Unload ASG-ZARA/AutoMedia DB using ZARAXTRT utility

Create Extract Data without HEADER and TRAILER Records (EDGRCZA1)

The EDGRCZA1 exec takes all records except the HEADER and TRAILER records and writes it into the output file. The additional sort step sorts the records by VOLUME, VOLUME SEQUENCE and FILE SEQUENCE.

This figure shows a sample JCL for EDGRCZA1 program:

```

//DELETE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE STSGWD.EDGJCA1.DATA NONVSAM PURGE
SET MAXCC = 0
/*
//*****
//* Create extract data without HEADER and TRAILER records *
//*****
//EXTRACT EXEC PGM=IKJEFT01,COND=(0,NE)
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*DETAIL DD DUMMY=*
//IN DD DISP=SHR,DSN=STSGWD.ZARA.DATA
//OUT DD DSN=&&EXTRACT,
// DISP=(,PASS),UNIT=SYSALLDA,
// SPACE=(TRK,(900,250),RLSE),
// DCB=(LRECL=370,RECFM=VB,BLKSIZE=0)
//SRTCNT DD DSN=&&CNTSTMT,
// DISP=(,PASS),UNIT=SYSALLDA,
// SPACE=(TRK,(1,1),RLSE),
// DCB=(LRECL=80,RECFM=F,BLKSIZE=0)
//SYSTSIN DD *
EX 'SYS1.SEDGEXE1(EDGRCZA1)'
/*

```

Figure 46. Sample JCL to Execute EDGRCZA1 (Part 1 of 4)

```

//*****
//* Sort data by VOLUME, VOLUME SEQUENCE and FILE SEQUENCE *
//*****
//SORT EXEC PGM=SORT,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//SORTIN DD DISP=(OLD,DELETE),DSN=&&EXTRACT
//SORTOUT DD DSN=STSGWD.EDGJCA1.DATA,
// DISP=(,CATLG,DELETE),UNIT=SYSDA,
// SPACE=(TRK,(900,250),RLSE),
// DCB=*.SORTIN

```

Figure 46. Sample JCL to Execute EDGRCZA1 (Part 2 of 4)

```

//SYSOUT DD SYSOUT=*
//* *****
//* *
//* * The EDGRCZA1 EXEC generates correct SORT statements *
//* * based on the FILEDATA offset in the VOLDATA *
//* * to use them in the SYSIN DD *
//* *
//* * ----- *
//* * the sort order of the output file must be: *
//* * 1. ascending first volume serial number in chain *
//* * 2. ascending volume sequence number *
//* * 3. ascending file sequence number *
//* * ----- *
//* *
//* * sort order for data set in FB format AutoMedia 1.3 *
//* * ----- *
//* * OPTION NOCHALT *
//* * SORT FIELDS=(9,6,CH,A,7,2,BI,A,164,2,BI,A) *
//* * OMIT COND=(1,6,CH,EQ,C'HEADER',OR, *
//* * 1,7,CH,EQ,C'TRAILER') *
//* *
//* * sort order for data set in FB format ASG-ZARA 1.4 & 1.5 *
//* * ----- *
//* * OPTION NOCHALT *
//* * SORT FIELDS=(9,6,CH,A,7,2,BI,A,170,2,BI,A) *
//* * OMIT COND=(1,6,CH,EQ,C'HEADER',OR, *
//* * 1,7,CH,EQ,C'TRAILER') *
//* *
//* * sort order for data set in FB format ASG-ZARA 1.6 & 1.7 *
//* * ----- *
//* * OPTION NOCHALT *
//* * SORT FIELDS=(9,6,CH,A,7,2,BI,A,176,2,BI,A) *
//* * OMIT COND=(1,6,CH,EQ,C'HEADER',OR, *
//* * 1,7,CH,EQ,C'TRAILER') *
//* *

```

Figure 46. Sample JCL to Execute EDGRCZA1 (Part 3 of 4)

```

/** * sort order for data set in VB format AutoMedia 1.3 *
/** * ----- *
/** * OPTION NOCHALT *
/** * SORT FIELDS=(13,6,CH,A,11,2,BI,A,168,2,BI,A) *
/** * OMIT COND=(5,6,CH,EQ,C'HEADER',OR, *
/** * 5,7,CH,EQ,C'TRAILER') *
/** * *
/** * sort order for data set in VB format ASG-ZARA 1.4 & 1.5 *
/** * ----- *
/** * OPTION NOCHALT *
/** * SORT FIELDS=(13,6,CH,A,11,2,BI,A,174,2,BI,A) *
/** * OMIT COND=(5,6,CH,EQ,C'HEADER',OR, *
/** * 5,7,CH,EQ,C'TRAILER') *
/** * *
/** * sort order for data set in VB format ASG-ZARA 1.6 & 1.7 *
/** * ----- *
/** * OPTION NOCHALT *
/** * SORT FIELDS=(13,6,CH,A,11,2,BI,A,180,2,BI,A) *
/** * OMIT COND=(5,6,CH,EQ,C'HEADER',OR, *
/** * 5,7,CH,EQ,C'TRAILER') *
/** * *
/** * *****
//SYSIN DD DISP=(OLD,DELETE),DSN=&&CNTSTMT

```

Figure 46. Sample JCL to Execute EDGRCZA1 (Part 4 of 4)

Input Data Sets

EDGRCZA1 has one input data set:

Table 63. EDGRCZA1 Input Data Sets

DDNAME	Description
IN	The unloaded ASG-ZARA/AutoMedia DB volume records.

Output Data Sets

EDGRCZA1 creates only one output data set which is sorted in a sort step by VOLUME, VOLUME SEQUENCE and FILE SEQUENCE. This data set is needed for the EDGRCZA2 extraction program.

Table 64. EDGRCZA1 Output Data Sets

DDNAME	Description
OUT	Contains all the created ASG-ZARA/AutoMedia DB records.

EDGRCZA1 has a special DD-statement to tailoring the output of messages. If this DD-statement is specified, (not commented out), additional messages will be written to the REPORT file:

Table 65. EDGRCZA1 special DETAIL DD

DDNAME	Description
DETAIL	To switch on/off additional message writing.

Return Codes

EDGRCZA1 sets a return code when it ends:

Table 66. EDGRCZA1 Return Codes

Return code	Description
0	Normal processing.
8	No input records found. Processing stops, no output records can be produced.
>8	An error has been encountered in the exec processing. Refer to the REXX error messages for resolution.

Messages

CZA1000I EDGRCZA1 STARTED *day month year hh mm ss*

Problem determination: where

day Day the function started.

month Month the function started.

year Year the function started.

hh mm ss

Time the function started.

Severity: Information

Explanation: None.

System action: None.

User response: None.

CZA1000I EDGRCZA1 ENDED *day month year hh mm ss* HIGHEST RETURN CODE WAS *ccc*

Problem determination: where

day Day the function started.

month Month the function started.

year Year the function started.

hh mm ss

Time the function started.

ccc Return code.

Severity: Information

Explanation: None.

System action: None.

User response: None.

CZA1001I YOU HAVE REQUESTED A ASG-ZARA/AUTOMEDIA *rrr* TO DFSMSRMM CONVERSION

Problem determination: where

rrr Version of your running ASG-ZARA/AUTOMEDIA system.

Severity: Information

Explanation: Automedia V1R3 or ZARA V1R4/V1R5 are supported.

System action: None.

User response: None.

CZA1002A PLEASE CHECK VOLUME *vol* BECAUSE VOLUME STATUS IS *volstat*

Problem determination: where

vol Tape volume serial number.

volstat Volume status.

Severity: Action

Explanation: Only volumes with a volume status of A (active), D (damaged) or S (scratch) will be converted. All other volumes are skipped.

System action: Processing continues.

User response: Check the volume status of this volume and correct it. Rerun the conversion process.

CZA1003I FIRST VOLSER IN CHAIN CHANGED FROM BLANK TO TAPE VOLUME *vol*

Problem determination: where

vol Tape volume serial number.

Severity: Information

Explanation: The first volses was blank and is changed to the current volume number for the next step.

System action: Processing continues.

User response: None.

CZA1004I FIRST VOLSER IN CHAIN CHANGED FROM *fool* TO TAPE VOLUME *vol*

Problem determination: where

fool Old first volume serial number.

vol New first volume serial number.

Severity: Information

Explanation: The first volses number was not correct for this kind of conversion and is changed to the current volume number for the next step.

System action: Processing continues.

User response: None.

CZA1005I PREV AND NEXT VOLUME SERIAL NUMBER IN CHAIN CLEARED FOR VOLSER *vol*

Problem determination: where

vol Volume serial number of processed record.

Severity: Information

Explanation: The previous and next volume fields were not empty. Both fields must be empty if the volume status is not active.

System action: None.

User response: None.

CZA1006E NO INPUT RECORDS FOUND

Severity: Error

Explanation: No valid records are available in the input file.

System action: Process stops, no output data created.

User response: Provide valid input data and restart the job.

CZA1007I YOU HAVE SWITCHED *ooo* DETAIL MESSAGE WRITING

Problem determination: where

ooo On or Off.

Severity: Information

Explanation: With the special DD-statement DETAIL, you have switched on/off additional message writing.

System action: None.

User response: None.

Convert Data out of the ASG-ZARA/AUTOMEDIA sorted Records (EDGRCZA2)

This section describes how to setup and control the ASG-ZARA/AutoMedia data conversion program EDGRCZA2. It converts the information out of the ASG-ZARA/AutoMedia records and creates temporary records for further processing.

The next figure shows a sample JCL for EDGRCZA2 program:

```

//DELETE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE STSGWD.EDGJCA2.DEXTRACT.DATA NONVSAM PURGE
DELETE STSGWD.EDGJCA2.DEXTOWN.DATA NONVSAM PURGE
DELETE STSGWD.EDGJCA2.DEXTVRS.DATA NONVSAM PURGE
DELETE STSGWD.EDGJCA2.CCARDS.DATA NONVSAM PURGE
DELETE STSGWD.EDGJCA2.ERRORS.DATA NONVSAM PURGE
DELETE STSGWD.EDGJCA2.REPORT.DATA NONVSAM PURGE
SET MAXCC = 0
/*
//*****
//* Create D-, L-, K-, and O-records. *
//*****
//EXTRACT EXEC PGM=IKJEFT01,COND=(0,NE)
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*DETAIL DD DUMMY
//VOLFILE DD DISP=SHR,DSN=STSGWD.EDGJCA1.DATA
//DEXTRACT DD DSN=STSGWD.EDGJCA2.DEXTRACT.DATA,
// DISP=(,CATLG,CATLG),UNIT=SYSALLDA,
// SPACE=(TRK,(900,250),RLSE),
// DCB=(RECFM=VB,LRECL=1024,BLKSIZE=0)
//DEXTOWN DD DSN=STSGWD.EDGJCA2.DEXTOWN.DATA,
// DISP=(,CATLG,CATLG),UNIT=SYSALLDA,
// SPACE=(TRK,(100,10),RLSE),
// DCB=(RECFM=VB,LRECL=1024,BLKSIZE=0)
//DEXTVRS DD DSN=STSGWD.EDGJCA2.DEXTVRS.DATA,
// DISP=(,CATLG,CATLG),UNIT=SYSALLDA,
// SPACE=(TRK,(900,250),RLSE),
// DCB=(RECFM=VB,LRECL=1024,BLKSIZE=0)
//CCARDS DD DSN=STSGWD.EDGJCA2.CCARDS.DATA,
// DISP=(,CATLG,CATLG),UNIT=SYSALLDA,
// SPACE=(TRK,(10,10),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//ERROR DD DSN=STSGWD.EDGJCA2.ERRORS.DATA,
// DISP=(,CATLG,CATLG),UNIT=SYSALLDA,
// SPACE=(TRK,(900,250),RLSE),
// DCB=(RECFM=FB,LRECL=133,BLKSIZE=0)
//REPORT DD DSN=STSGWD.EDGJCA2.REPORT,
// DISP=(,CATLG,CATLG),UNIT=SYSALLDA,
// SPACE=(TRK,(900,250),RLSE),
// DCB=(RECFM=FB,LRECL=133,BLKSIZE=0)

```

Figure 47. Sample JCL to Execute EDGRCZA2 (Part 1 of 2)

```

/** ----- *
/** You can use the CUSTOM DD-statement to: *
/** *
/** 1. exclude volumes from the EDGRCZA2 extraction process. *
/** 2. set a default MV, if no other MV matched *
/** 3. specify the default location for all MV VRS definitions *
/**    one of HOME, SHELF or CURRENT *
/** 4. specify the default conversion owner if no owner is found *
/** *
/** SYNTAX: IF fieldname EQUALS value THEN SKIP or MODE value *
/** *
/** Possible keywords and values in this version: *
/** *
/** VOLCREAT, VOLSTAT1, DATE, MVDEF, MVLOC and OWNER *
/** *
/** ----- *
//CUSTOM DD *
/*IF VOLCREAT EQUALS R THEN SCRATCH
/*IF VOLSTAT1 EQUALS D THEN SKIP
IF DATE EQUALS ' ' THEN SET 1960000C
IF MVDEF EQUALS ' ' THEN CATALOG
IF MVLOC EQUALS ' ' THEN HOME
IF OWNER EQUALS ' ' THEN ZARACNVT
//SYSYSIN DD *
      EX 'SYS1.SEDGEXE1(EDGRCZA2)' 'NEW'
/*

```

Figure 47. Sample JCL to Execute EDGRCZA2 (Part 2 of 2)

Input Data Sets

EDGRCZA2 has one input data set:

Table 67. EDGRCZA2 Input Data Sets

DDNAME	Description
VOLFILE	Contains sorted ASG-ZARA/AutoMedia DB records.

Output Data Sets

EDGRCZA2 creates 6 different output data sets. These data sets are needed as input to the EDGCNVT conversion program.

Table 68. EDGRCZA2 Output Data Sets

DDNAME	Description
CCARDS	Contains all created EDGCNVT SYSIN file statements that can be used for EDGCNVT processing.
DEXTRACT	Contains all created library and data set records for input to the EDGCNVT conversion program.
DEXTOWN	Contains all created owner records for input to the EDGCNVT conversion program
DEXTVRS	Contains all created vital record specification records for input to the EDGCNVT conversion program.
ERROR	Contains warning and error messages associated with the data extraction process.
REPORT	Contains informational messages associated with the data extraction process.

EDGRCZA2 has a special DD-statement to tailoring the output of messages. If this DD-statement is specified, (not commented out), additional messages will be written to the REPORT file:

Table 69. EDGRCZA2 special DETAIL DD

DDNAME	Description
DETAIL	To switch on/off additional message writing.

Return Codes

EDGRCZA2 sets a return code when it ends:

Table 70. EDGRCZA2 Return Codes

Return code	Description
0	Normal processing.
4	One or more warning messages issued. Valid records for input to EDGCNVT are produced, but the warning messages should be reviewed to ensure that results are as expected. See file REPORT.
8	No input records found. Processing stops, no output records can be produced.
12	Data error, see file ERROR. Processing stops, no output records can be produced.

Messages

CZA2000I EDGRCZA2 STARTED *day month year hh mm ss*

Problem determination: where

day Day the function started.

month Month the function started.

year Year the function started.

hh mm ss
Time the function started.

Severity: Information

Explanation: None.

System action: None.

User response: None.

CZA2000I EDGRCZA2 ENDED *day month year hh mm ss* HIGHEST RETURN CODE WAS *ccc*

Problem determination: where

day Day the function started.

month Month the function started.

year Year the function started.

hh mm ss

Time the function started.

ccc Return code.

Severity: Information

Explanation: None.

System action: None.

User response: None.

CZA2001I YOU HAVE SWITCHED *ooo* DETAIL MESSAGE WRITING

Problem determination: where

ooo On or Off.

Severity: Information

Explanation: With the special DD-statement *DETAIL*, you have switched on/off additional message writing.

System action: None.

User response: None.

CZA2002I YOU HAVE USED A CUSTOM DD-statement

Severity: Information

Explanation: A CUSTOM DD-statement is used in the job control to tailor this run.

System action: None.

User response: None.

CZA2002I YOU HAVE NOT USED A CUSTOM DD STATEMENT

Severity: Information

Explanation: No CUSTOM DD-statement is used in the job control to tailor this run. The default values are used.

System action: None.

User response: None.

CZA2003I FIRST VOLUME *fvol*, AND NEXT VOLUME *nvool* ARE PART OF A MV/MF CHAIN. FILE SEQ. *fseq*

Problem determination: where

fvol Volume serial number of the first volume in the chain.

nvool Volume serial number of the next volume in the chain.

fseq File sequence number of the current processed file.

Severity: Information

Explanation: Information that the file expands more than one volume.

System action: None.

User response: None.

CZA2004E PREVIOUS VOLUME *pppppp* OF VOLUME *vvvvvv* STATUS *ssssss* HAS ANOTHER OR NO NEXT VOLUME.

Problem determination: where

pppppp Previous volume serial number in the chain.

vvvvvv Volume serial number of the current processed record.

sssss Status of the current processed volume.

Severity: Error

Explanation: The previous volume or the current processed volume has invalid information.

System action: The processing stops with return code = 12.

User response: Check the record and correct the error. Rerun the job.

CZA2005E NO INPUT RECORDS FOUND

Severity: Error

Explanation: No valid records are available in the input file.

System action: Process stops, no output data created.

User response: Provide valid input data and restart the job.

CZA2006I THE CHECK-IN DATE *dddddd* FOR VOLUME *vvvvvv* NOT USED

Problem determination: where

dddddd The volume check-in date.

vvvvvv Volume serial number of the current volume record.

Severity: Information

Explanation: The volume check-in date is blank. The default date will be used.

System action: Processing continues.

User response: None.

CZA2006I THE DATE CHECK-IN *dddddd* FOR VOLUME *vvvvvv* NOT USED

Problem determination: where

dddddd The volume check-in date.

vvvvvv Volume serial number of the current volume record.

Severity: Information

Explanation: The volume check-in date is blank. The default date will be used.

System action: Processing continues.

User response: None.

CZA2007I THE VOLUME MOVE DATE *mmmmmm* FOR VOLUME *vvvvvv* NOT USED

Problem determination: where

mmmmmm

The volume move date.

vvvvvv Volume serial number of the current volume record.

Severity: Information

Explanation: The volume move date is blank.

System action: The default date is used.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2008I THE CLEAN DATE *cccccc* FOR VOLUME *vvvvvv* NOT USED

Problem determination: where

cccccc The volume clean date.

vvvvvv Volume serial number of the current volume record.

Severity: Information

Explanation: The volume clean date is blank.

System action: The default date is used.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2009I THE FIRST USED DATE *ffffff* FOR VOLUME *vvvvvv* NOT USED

Problem determination: where

ffffff The first used date.

vvvvvv Volume serial number of the current volume record.

Severity: Information

Explanation: The volume first used date is blank. The default date will be used.

System action: Processing continues.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2010I THE WIPED DATE *wwwwww* FOR VOLUME *vvvvvv* NOT USED

Problem determination: where

wwwwww
The wiped date.

vvvvvv Volume serial number of the current volume record.

Severity: Information

Explanation: The wiped date is blank. The default date will be used.

System action: Processing continues.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2011I THE EXPIRATION DATE *eeeeee* FOR DSN *dddddd* ON VOLUME *vvvvvv* NOT USED *defdate*

Problem determination: where

eeeeee The expiration date.

dddddd Data set name.

vvvvvv Volume serial number of the current volume record.

defdate Default expiration date used

Severity: Information

Explanation: The expiration date is blank.

System action: The default date is used.

User response: None.

CZA2012W THE NO OF COPIES/CYCLES FIELD *nnnnnn* ON VOLUME *vvvvvv* WAS BLANK AND IS SET TO HEX 0000

Problem determination: where

nnnnnn The number of copies/cycles.

vvvvvv Volume serial number of the current volume record.

Severity: Warning

Explanation: The field "No of copies/cycles" is blank.

System action: The number of copies/cycles field is set to hex zero.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2013W THE FILECNTX VALUE *ffffff* IS REDUCED TO 999 FOR VOLUME *vvvvvv*

Problem determination: where

ffffff The number of copies/cycles.

vvvvvv Volume serial number of the current volume record.

Severity: Warning

Explanation: The current processed volume record has a file count greater 999.

System action: The number of copies/cycles is reduced to maximal allowed number of 999. The process continues with return code = 4.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2014W THE MV IS CHANGED FROM 000 TO *ffffff* FOR VOLUME *vvvvvv*

Problem determination: where

ffffff The number of copies or cycles.

vvvvvv Volume serial number of the current volume record.

Severity: Warning

Explanation: The current processed volume record has an empty management value MV field.

System action: The management value field MV is set to the number of cycles/copies used. Processing continues with return code 4.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2015W OWNER *000000* HAS MORE THAN 8 CHARS

Problem determination: where

000000 Owner id.

Severity: Warning

Explanation: The owner information has more then 8 characters.

System action: The default owner id is used. Processing continues with return code = 4.

User response: Correct the error and rerun the job. See message CZA2016I.

CZA2016I OWNER *000000* WILL BE USED

Problem determination: where

000000 Default owner id.

Severity: Information

Explanation: The content of the owner field was wrong.

System action: The default owner ID is used.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2017I THE CREATION DATE dddddd FOR VOLUME vvvvvv NOT USED

Problem determination: where

ddddd Creation date.

vvvvv Volume serial number of the current volume record.

Severity: Information

Explanation: The current processed volume record has an incorrect creation date format.

System action: The default date is used.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2018I THE EXPIRATION DATE eeeee FOR VOLUME vvvvvv NOT USED

Problem determination: where

eeeeee Volume expiration date.

vvvvv Volume serial number of the current volume record.

Severity: Information

Explanation: The current processed volume record has an incorrect expiration date format in the data set section. FILDATEX was the checked field.

System action: The default date is used.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2019W VOLUME STATUS CHANGED TO SCRATCH FROM STATUS ssssss FOR VOLUME vvvvvv

Problem determination: where

sssss Current volume status deleted or damaged.

vvvvv Volume serial number of the current volume record.

Severity: Warning

Explanation: The status of the current processed volume record has been changed, because you have requested this change via CUSTOM DD-statement.

System action: The status is changed to SCRATCH. Processing continues.

User response: None.

CZA2020W COUNT OF DS ON VOLUME vvvvvv WITH STATUS ssssss CHANGED TO 1

Problem determination: where

vvvvv Volume serial number of the current volume record.

sssss Current volume status.

Severity: Warning

Explanation: The number of data sets for the current processed volume has been changed to a value of one.

System action: Data set count changed to one. Processing continues with return code = 4.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2021I THE DATE LAST USED dddddd FOR VOLUME vvvvvv NOT USED

Problem determination: where

dddddd Date last used.

vvvvvv Volume serial number of the current volume record.

Severity: Information

Explanation: The current processed volume record has an incorrect last used date format.

System action: The default date is used.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2022I THE DSN CREATE DATE dddddd FOR VOLUME vvvvvv NOT USED

Problem determination: where

dddddd DSN create date.

vvvvvv Volume serial number of the current volume record.

Severity: Information

Explanation: The current processed volume record has an incorrect DSN create date format.

System action: The default date is used.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2023W THE DSN CREATE DATE IS CHANGED TO 1999/300

Severity: Warning

Explanation: The current processed volume record has an incorrect DSN create date format and the date is changed to 1999/300.

System action: The hard coded date 1999/300 is used.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process, see message CZA2022I.

CZA2024W THE ASSIGN TIME tttttt uuuuuu FOR VOLUME vvvvvv CORRECTED

Problem determination: where

tttttt Creation time.

uuuuuu Last used time.

vvvvvv Volume serial number of the current volume record.

Severity: Warning

Explanation: The current processed volume record has an incorrect creation time format.

System action: The creation time in DFSMSrmm called assign time is set to hex zero.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2025W THE LRECL llllll FOR VOLUME vvvvvv SET TO 0

Problem determination: where

llllll Current record length.

vvvvvv Volume serial number of the current volume record.

Severity: Warning

Explanation: The current processed volume record has an incorrect LRECL format.

System action: The LRECL is set to hex zero.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2026W THE BLKSIZE *bbbbbb* FOR VOLUME *vvvvvv* SET TO *nnnnnn*

Problem determination: where

bbbbbb Current block size.

vvvvvv Volume serial number of the current volume record.

nnnnnn The new block size.

Severity: Warning

Explanation: The current processed volume record has an incorrect BLKSIZE format.

System action: The BLKSIZE is set to hex zero. The process continues with return code = 4.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2027I THE BLOCK SIZE FOR VOLUME *vvvvvv* IS *bbbbbb*

Problem determination: where

vvvvvv Volume serial number of the current volume record.

bbbbbb Current block size.

Severity: Information

Explanation: The current block size for volume.

System action: None.

User response: None.

CZA2028W THE BLOCK COUNT *bbbbbb* FOR VOLUME *vvvvvv* SET TO *sssss*

Problem determination: where

bbbbbb Current block count.

vvvvvv Volume serial number of the current volume record.

sssss The new block count.

Severity: Warning

Explanation: The current processed volume record has an incorrect block count.

System action: The block count is set to hex zero. Processing continues with return code = 4.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2029I THE CALCULATED K-BYTES FOR VOLUME *vvvvvv* IS *kkkkkk*

Problem determination: where

vvvvvv Volume serial number of the current volume record.

kkkkkk The calculated bytes in kilo.

Severity: Information

Explanation: The calculated bytes in kilo.

System action: None.

User response: None.

CZA2031W THE CREATION TIME cccccc uuuuuu FOR VOLUME vvvvvv CORRECTED

Problem determination: where

ccccc Creation time.

uuuuuu Last used time.

vvvvvv Volume serial number of the current volume record.

Severity: Warning

Explanation: The current processed volume record has an incorrect creation time format.

System action: The creation time is corrected. Processing continues with return code = 4.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2032W THE CREATION TIME cccccc uuuuuu FOR DATASET dddddd ON VOLUME vvvvvv

Problem determination: where

ccccc Creation time.

uuuuuu Last used time.

dddddd Data set name.

vvvvvv Volume serial number of the current volume record.

Severity: Warning

Explanation: The current processed volume record has an incorrect creation time format in the FILDATA section.

System action: The data set creation time is corrected. Processing continues with return code = 4.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2033W DSN SET IN FUNCTION fff. DSN dddddd ADDED TO VOLUME vvvvvv

Problem determination: where

fff Internal subroutine used at this time, it can be UDSN or UVOL.

dddddd Data set name.

vvvvvv Volume serial number of the current volume record.

Severity: Information

Explanation: A data set name on the current processed volume record is missing.

System action: The default data set name "DSN.SET.BY.CONVERT" is added.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2034I THE VRS rrrrrr IS ALREADY DEFINED

Problem determination: where

rrrrrr Vital record specification (VRS).

Severity: Information

Explanation: The current processed vital record specification is already added at a previous process.

System action: None.

User response: None.

CZA2035I SKIP THE CREATION OF THE VRS RECORD

Severity: Information

Explanation: The VRS is already defined , the new creation is skipped.

System action: None.

User response: None.

CZA2036I VITAL-RECORDS RECORD FOR DSN *dddddd* CREATED

Problem determination: where

dddddd Data set name.

Severity: Information

Explanation: A data set name vital record specification (VRS) is created.

System action: None.

User response: None.

CZA2037I THE DEFAULT DATE USED FOR ALL BLANK DATE FIELDS IS *dddddd*

Problem determination: where

dddddd The default date.

Severity: Information

Explanation: Tells you the default date used if a date field is blank.

System action: None.

User response: None.

CZA2038I THE LOCATION *//////* IS ALREADY DEFINED

Problem determination: where

////// Location name.

Severity: Information

Explanation: The current processed location is already added.

System action: None.

User response: None.

CZA2039I SKIP THE CREATION OF THE LOCATION INFORMATION

Severity: Information

Explanation: The location is already defined, it is skipped.

System action: None.

User response: None.

CZA2040I LIBRARY-RECORD FOR LIB *//////* CREATED

Problem determination: where

////// Current location.

Severity: Information

Explanation: A library record for the current processed location is added to CCARDS output file.

System action: None.

User response: None.

CZA2041I THE OWNER 000000 IS ALREADY DEFINED

Problem determination: where

000000 Current processed owner.

Severity: Information

Explanation: The current processed owner is already added at a previous process.

System action: None.

User response: None.

CZA2042I SKIP THE CREATION OF THE OWNER RECORD

Severity: Information

Explanation: The owner is already defined, the new creation is skipped.

System action: None.

User response: None.

CZA2043I O-RECORD FOR OWNER 000000 CREATED

Problem determination: where

000000 Current processed owner.

Severity: Information

Explanation: The current processed owner is added to the DEXTOWN output file.

System action: None.

User response: None.

CZA2044I SET THE DEFAULT MANAGEMENT VALUE mmmmmmm IF MV IS BLANK

Problem determination: where

mmmmmm

Name of management value.

Severity: Information

Explanation: You have specified via CUSTOM DD-statement that a default management value is set for all data sets that have no management value.

System action: None.

User response: None.

CZA2045I SET THE DEFAULT LOCATION FOR ALL MANAGEMENT VALUES TO llllll

Problem determination: where

llllll Location name.

Severity: Information

Explanation: You have specified via CUSTOM DD-statement that a default location is set for all volumes that have no location information.

System action: None.

User response: None.

CZA2046I ALL VOLUMES WITH A CREATOR INFO OF cccccc WILL BE EXCLUDED

Problem determination: where

ccccc ASG-ZARA/AUTOMEDIA creator information.

Severity: Information

Explanation: You have requested via CUSTOM DD-statement that all volumes with a specified ASG-ZARA/AutoMedia creator information should be excluded in this run.

System action: None.

User response: None.

CZA2047I ALL VOLUMES WITH A STATUS OF ssssss WILL BE EXCLUDED

Problem determination: where

ssssss ASG-ZARA/AUTOMEDIA volume status.

Severity: Information

Explanation: You have requested via CUSTOM DD-statement that all volumes with a specified ASG-ZARA/AutoMedia volume status information should be excluded in this run.

System action: None.

User response: None.

CZA2048E WRONG CUSTOM CONTROL CARD cccccc

Problem determination: where

ccccc Control card.

Severity: Error

Explanation: Syntax error in the CUSTOM DD-statement input file.

System action: The process ends with return code = 12.

User response: Correct the CUSTOM DD-statement input and rerun the job.

CZA2049I ALL VOLUMES WITH A STATUS OF "DAMAGED (HAS PERM I/O ERROR)", WILL BE CHANGED TO STATUS "SCRATCH" AND THE ERROR COUNT IS SET TO ZERO

Severity: Information

Explanation: You have requested via CUSTOM DD-statement that all volumes with a specified ASG-ZARA/AutoMedia volume status information of damaged should be changed to SCRATCH and the error count is cleared.

System action: None.

User response: None.

CZA2050I THE CREATION OF D-RECORDS IS SWITCH TO scrdsn

Problem determination: where

scrdsn Specifies the kind of extraction processing. it can be old or new.

Severity: Information

Explanation: The EDGRCZA2 REXX exec is designed to support the OLD and the NEW version of the EDGCNVT conversion program. If you have specified OLD only the first file information for a scratch volume will be converted. If you have specified NEW (default) all file information are converted.

System action: None.

User response: None.

CZA2051I DEFAULT OWNER USED IN THIS RUN 000000

Problem determination: where

000000 Default owner id.

Severity: Information

Explanation: The owner you have specified is used if the owner field is blank or does not match the DFSMSrmm naming conventions owner ID's.

System action: Overwrite the hard coded default owner ID ZARACNVT.

User response: None.

CZA2052I DEFAULT DATE USED FOR ALL BLANK DATE FIELDS IS dddddd

Problem determination: where

dddddd Default date.

Severity: Information

Explanation: The default date you have specified is used if one of the date fields is blank or does not match the DFSMSrmm date conventions.

System action: Overwrite the hard coded default date 1999/300.

User response: None.

CZA2060I DATASET RECORDING FOR VOLUME vvvvvv SWITCHED OFF

Problem determination: where

vvvvvv Current volume serial number.

Severity: Information

Explanation: The data set recording for the current processed volume is switched off, because the ASG-ZARA/AutoMedia VOLCREAT information is 'keep only first file information (report distribution)'.

System action: Data set recording is switched off in FLAGD.

User response: None.

CZA2071I VOLUME vvvvvv EXCLUDED BECAUSE VOLCREAT = N

Problem determination: where

vvvvvv Current volume serial number.

Severity: Information

Explanation: You have requested via CUSTOM DD-statement that all volumes with a volume creator information of 'Volume created by open, close, EOVS processing' should be excluded in this run.

System action: Record skipped.

User response: None.

CZA2072I VOLUME vvvvvv EXCLUDED BECAUSE VOLCREAT = I

Problem determination: where

vvvvvv Current volume serial number.

Severity: Information

Explanation: You have requested via CUSTOM DD-statement that all volumes with a volume creator information of 'Volume range-defined and initialized' should be excluded in this run.

System action: Record skipped.

User response: None.

CZA2073I VOLUME *vvvvvv* EXCLUDED BECAUSE VOLSTAT1 = R

Problem determination: where

vvvvvv Current volume serial number.

Severity: Information

Explanation: You have requested via CUSTOM DD-statement that all volumes with a volume creator information of 'Volume range-defined, but unused' should be excluded in this run.

System action: Record skipped.

User response: None.

CZA2074I VOLUME *vvvvvv* EXCLUDED BECAUSE VOLCREAT = C

Problem determination: where

vvvvvv Current volume serial number.

Severity: Information

Explanation: You have requested via CUSTOM DD-statement that all volumes with a volume creator information of 'Volume created by conversion' should be excluded in this run.

System action: Record skipped.

User response: None.

CZA2075I VOLUME *vvvvvv* EXCLUDED BECAUSE VOLCREAT = L

Problem determination: where

vvvvvv Current volume serial number.

Severity: Information

Explanation: You have requested via CUSTOM DD-statement that all volumes with a volume creator information of 'Volume has no inter-volume linkage (HSM)' should be excluded in this run.

System action: Record skipped.

User response: None.

CZA2076I VOLUME *vvvvvv* EXCLUDED BECAUSE VOLCREAT = F

Problem determination: where

vvvvvv Current volume serial number.

Severity: Information

Explanation: You have requested via CUSTOM DD-statement that all volumes with a volume creator information of 'Keep only first file information (report distribution)' should be excluded in this run.

System action: Record skipped.

User response: None.

CZA2077I VOLUME *vvvvvv* EXCLUDED BECAUSE VOLCREAT = B

Problem determination: where

vvvvvv Current volume serial number.

Severity: Information

Explanation: You have requested via CUSTOM DD-statement that all volumes with a volume status of 'Keep only

first file information, no inter-volume linkage' should be excluded in this run.

System action: Record skipped.

User response: None.

CZA2078I VOLUME *vvvvvv* EXCLUDED BECAUSE VOLSTAT1 = A

Problem determination: where

vvvvvv Current volume serial number.

Severity: Information

Explanation: You have requested via CUSTOM DD-statement that all volumes with a volume status of 'Volume is active' should be excluded in this run.

System action: Record skipped.

User response: None.

CZA2079I VOLUME *vvvvvv* EXCLUDED BECAUSE VOLSTAT1 = S

Problem determination: where

vvvvvv Current volume serial number.

Severity: Information

Explanation: You have requested via CUSTOM DD-statement that all volumes with a volume status of 'Volume can be used as scratch' should be excluded in this run.

System action: Record skipped.

User response: None.

CZA2080I VOLUME *vvvvvv* EXCLUDED BECAUSE VOLSTAT1 = D

Problem determination: where

vvvvvv Current volume serial number.

Severity: Information

Explanation: You have requested via CUSTOM DD-statement that all volumes with a volume status of 'Volume is deleted' should be excluded in this run.

System action: Record skipped.

User response: None.

CZA2081I VOLUME *vvvvvv* EXCLUDED BECAUSE VOLSTAT1 = E

Problem determination: where

vvvvvv Current volume serial number.

Severity: Information

Explanation: You have requested via CUSTOM DD-statement that all volumes with a volume status of 'Volume is damaged (permanent I/O error)' should be excluded in this run.

System action: Record skipped.

User response: None.

CZA2082I THE CREATION TIME *ttttt* *uuuuuu* FOR VOLUME *vvvvvv*

Problem determination: where

ttttt Creation time.

uuuuuu Last used time.

vvvvvv Current volume serial number.

Severity: Information

Explanation: Displays the creation time of the current processed volume.

System action: None.

User response: None.

CZA2083I THE BLOCK COUNT FOR VOLUME *vvvvvv* IS *bbbbbb*

Problem determination: where

vvvvvv Current volume serial number.

bbbbbb Block count.

Severity: Information

Explanation: Displays the block count of the current processed volume.

System action: None.

User response: None.

CZA2084I THE CREATION TIME *ttttt* *uuuuuu* FOR DATASET *dddddd* ON VOLUME *vvvvvv*

Problem determination: where

ttttt Creation time.

uuuuuu Last used time.

dddddd Data set name.

vvvvvv Current volume serial number.

Severity: Information

Explanation: Displays the data set creation time of the current processed volume record.

System action: None.

User response: None.

CZA2085I DSN= *dddddd* LOC= *lllll* CYCLES= *ccccc*

Problem determination: where

dddddd Data set name.

lllll Location name.

ccccc Number of cycles or copies.

Severity: Information

Explanation: A data set name vital record specification record is created.

System action: Record added to file DEXTVRS.

User response: None.

CZA2086I KREC: VDATE >>>nnnnnn<<< DATATYPE(xx)

Problem determination: where

nnnnnn Name.

xx Number of cycles or copies.

Severity: Information

Explanation: A name vital record specification record is created.

System action: Record added to file DEXTVRS.

User response: None.

CZA2087I KREC VDATE >>>nnnnnn<<<

Problem determination: where

nnnnnn Name.

Severity: Information

Explanation: The number of cycles or copies changed to hex zero, because the number processed was not corrected.

System action: Number of cycles or copies changed to hex zero.

User response: See message CZA2086 for more information.

CZA2099E INVALID SPECIAL EXPDT dddddd FOUND ON VOLUME vvvvvv

Problem determination: where

dddddd Expiration date.

vvvvvv Volume serial number.

Severity: Error

Explanation: A special expiration date of 1998/000 was found. This expiration date is used for bypass tape management function.

System action: Skip the creation of a name vital record specification. The process ends with return code = 12.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2116I DEFAULT OWNER 000000 WILL BE USED

Problem determination: where

000000 Owner id.

Severity: Information

Explanation: The current processed volume record has no owner information or the owner information does not match MVS naming conventions.

System action: The ASG-ZARA/AutoMedia owner is replaced by the default owner.

User response: None.

CZA2117W OWNER 000000 HAS ONE OR MORE INVALID CHARS

Problem determination: where

000000 Owner id.

Severity: Warning

Explanation: The current processed owner information has one or more invalid character(s). An owner must follow the MVS standard naming conventions.

System action: The default owner is used. The process continues with return code = 4.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA2118W OWNER 000000 HAS AN INVALID FIRST CHARACTER

Problem determination: where

000000 Owner id.

Severity: Warning

Explanation: The current processed owner information is not starting with a character or national character.

System action: The default owner is used. The process continues with return code = 4.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

Create VRS records from the ASG-ZARA/AUTOMEDIA Expiration Report (EDGRCZA3)

This section describes how to setup and control the ASG-ZARA/AutoMedia expiration report extraction program.

The following figure shows a sample JCL for EDGRCZA3 program:

```

//DELETE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE STSGWD.EDGJCA3.DEXTVRS.DATA NONVSAM PURGE
DELETE STSGWD.EDGJCA3.REPORT.DATA NONVSAM PURGE
DELETE STSGWD.EDGJCA3.ERRORS.DATA NONVSAM PURGE
SET MAXCC = 0
/*
//*****
//* Create K-records. *
//*****
//EXTRACT EXEC PGM=IKJEFT01,COND=(0,NE)
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*DETAIL DD DUMMY
//ZARAEDCD DD DISP=SHR,DSN=STSGWD.ZARA.EXPIRE.REPORT
//DEXTVRS DD DSN=STSGWD.EDGJCA3.DEXTVRS.DATA,
// DISP=(,CATLG,CATLG),UNIT=SYSALLDA,
// SPACE=(TRK,(900,250),RLSE),
// DCB=(RECFM=VB,LRECL=1024,BLKSIZE=0)
//*MERGED DD DSN=STSGWD.EDGJCA3.MERGED.DATA,
//* DISP=(,CATLG,CATLG),UNIT=SYSALLDA,
//* SPACE=(TRK,(900,250),RLSE),
//* DCB=(RECFM=VB,LRECL=1024,BLKSIZE=0)
//ERROR DD DSN=STSGWD.EDGJCA3.ERRORS.DATA,
// DISP=(,CATLG,CATLG),UNIT=SYSALLDA,
// SPACE=(TRK,(900,250),RLSE),
// DCB=(RECFM=FB,LRECL=163,BLKSIZE=0)
//REPORT DD DSN=STSGWD.EDGJCA3.REPORT.DATA,
// DISP=(,CATLG,CATLG),UNIT=SYSALLDA,
// SPACE=(TRK,(900,250),RLSE),
// DCB=(RECFM=FB,LRECL=133,BLKSIZE=0)
//SYSTSIN DD *
EX 'SYS1.SEDGEXE1(EDGRCZA3)'
/*
//*****
//* Sort created K-records and remove duplicate *
//*****
//SORT EXEC PGM=SORT
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=STSGWD.EDGJCA3.DEXTVRS.DATA
//SORTOUT DD DISP=SHR,DSN=STSGWD.EDGJCA3.DEXTVRS.DATA
//SYSIN DD *
OPTION NOCHALT
SORT FIELDS=(5,180,CH,A),DYNALLOC
SUM FIELDS=NONE removes duplicates
END
/*

```

Figure 48. Sample JCL to Execute EDGRCZA3

Input Data Sets

EDGRCZA3 has one input data set:

Table 71. EDGRCZA3 Input Data Sets

DDNAME	Description
ZARAEDCD	Contains ASG-ZARA/AutoMedia expiration report.

Output Data Sets

EDGRCZA3 creates three output data set:

Table 72. EDGRCZA3 Output Data Sets

DDNAME	Description
DEXTVRS	Contains all created vital records for input to the EDGCNVT conversion program.
ERROR	Contains warning and error messages associated with the data extraction process.
REPORT	Contains informational messages associated with the data extraction process.

EDGRCZA3 has a special DD-statement to tailoring the output of messages. If this DD-statement is specified, (not commented out), additional messages will be written to the REPORT file:

Table 73. EDGRCZA3 special DETAIL DD

DDNAME	Description
DETAIL	To switch on/off additional message writing.

Return Codes

EDGRCZA3 sets a return code when it ends:

Table 74. EDGRCZA3 Return Codes

Return code	Description
0	Normal processing.
4	One or more warning messages issued. Valid records for input to EDGCNVT are produced, but the warning messages should be reviewed to ensure that results are as expected.
8	No input records found. Processing stops, no output records can be produced.
12	Data error, see file ERROR. Processing stops, no output records can be produced.

Data set name mask and jobname mask correcting

Until the conversion, we have to check the data set name for each expiration date candidate definition.

The following will be checked:

1. The low level qualifier of the data set name has 9 characters ending with an asterisk. The data set name is changed to ".*".
2. Length of each qualifier of the data set name of the length. It must be eight or less characters.
3. Each data set name qualifier includes only valid characters (alphanumeric characters, \$,# or @).
4. The first byte of each qualifier of the data set name cannot be a number.
5. If the last character is an asterisk, then change the data set name mask to ".*.*".
6. If data set name ends with ".*", then the data set name is changed to ".*".
7. For all other data set name masks we will create a PSEUDO GDG vital record definition too (G^V^V^V^V^V^V^V^V^V^V).

In ASG-ZARA/AutoMedia the data set name must not have any special character at the end to match to a full specific data set name or to a GDG.

All question marks (?) are translated to percent sign (%).

Messages

CZA3000I EDGRCZA3 STARTED *day month year hh mm ss*

Problem determination: where

day Day the function started.

month Month the function started.

year Year the function started.

hh mm ss
Time the function started.

Severity: Information

Explanation: None.

System action: None.

User response: None.

CZA3000I EDGRCZA3 ENDED *day month year hh mm ss* HIGHEST RETURN CODE WAS *ccc*

Problem determination: where

day Day the function ended.

month Month the function ended.

year Year the function ended.

hh mm ss
Time the function ended.

ccc Return code.

Severity: Information

Explanation: None.

System action: None.

User response: None.

CZA3001I YOU HAVE SWITCHED *ooo* DETAIL MESSAGE WRITING

Problem determination: where

ooo On or off.

Severity: Information

Explanation: With the special DD-statement DETAIL, you have switched on/off additional message writing.

System action: None.

User response: None.

CZA3002I YOU HAVE SWITCHED *ooo* CREATION OF MERGED EXPIRATION AND VAULT RECORDS

Problem determination: where

ooo On or off.

Severity: Information

Explanation: With the special DD-statement MERGED, you have switched on/off migration of vault and retention records.

System action: None.

User response: None.

CZA3010W THE COUNT FOR *dddddd* CHANGED TO A VALUE OF 99999

Problem determination: where

dddddd Current processed data set record.

Severity: Warning

Explanation: The current input record does not have 1 whole number between 1 and 99999 in the count field.

System action: Record corrected and the process continues with return code = 4.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA3011W THE COUNT1 FOR *dddddd* CHANGED TO A VALUE OF 99999

Problem determination: where

dddddd Current processed data set record.

Severity: Warning

Explanation: The current input record does not have 1 whole number between 1 and 99999 in the count field.

System action: Record corrected and the process continues with return code = 4.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA3012E NO INPUT RECORD FOUND

Severity: Error

Explanation: The data set specified in ZARAEDCD DD-statement was empty.

System action: Process ends with return code = 8.

User response: Check the ASG-ZARA/AutoMedia input file, correct the error and rerun the conversion process.

CZA3050I THE FOLLOWING RETENTION TYPE IS USED >>*rrrrrr*<<

Problem determination: where

rrrrrr Current retention type.

Severity: Information

Explanation: The retention date used. It can be one of blank or // to use the JCL retention, NE or PE for permanent retention, CA for catalog retention, LU for last used retention, US for user defined retention, or CY for cycle controlled retention.

System action: None.

User response: None.

**CZA3051I DATASET NAME PATTERN: *pppppp* ASG-ZARA/AUTOMEDIA RETENTION TYPE DATE
DFSMSRMM RETENTION TYPE DAYS COUNT(99999) DELETEDATE(*yymmdd*)**

Problem determination: where

pppppp Data set name pattern used.

yymmdd

VRS retention date.

Severity: Information

Explanation: This message shows the data set name pattern is used to create a VRS record.

System action: None.

User response: None.

**CZA3052I DATASET NAME PATTERN: *pppppp* ASG-ZARA/AUTOMEDIA RETENTION TYPE DATE
DFSMSRMM RETENTION TYPE DAYS COUNT(99999) DELETEDATE(*yymmdd*)**

Problem determination: where

pppppp Data set name pattern used.

yymmdd

VRS retention date.

Severity: Information

Explanation: This message shows the data set name pattern is used to create a VRS record.

System action: None.

User response: None.

**CZA3053I DATASET NAME PATTERN: *pppppp* ASG-ZARA/AUTOMEDIA RETENTION TYPE NEVER
DFSMSRMM RETENTION TYPE DAYS COUNT(99999)**

Problem determination: where

pppppp Data set name pattern used.

Severity: Information

Explanation: This message shows the data set name pattern is used to create a VRS record.

System action: None.

User response: None.

**CZA3054I DATASET NAME PATTERN: *pppppp* ASG-ZARA/AUTOMEDIA RETENTION TYPE PERM
DFSMSRMM RETENTION TYPE DAYS COUNT(99999)**

Problem determination: where

pppppp Data set name pattern used.

Severity: Information

Explanation: This message shows the data set name pattern is used to create a VRS record.

System action: None.

User response: None.

**CZA3055I DATASET NAME PATTERN: *pppppp* ASG-ZARA/AUTOMEDIA RETENTION TYPE CATALOG
DFSMSRMM RETENTION TYPE DAYS COUNT(99999) WHILECATALOG**

Problem determination: where

pppppp Data set name pattern used.

Severity: Information

Explanation: This message shows the data set name pattern is used to create a VRS record.

System action: None.

User response: None.

CZA3056I DATASET NAME PATTERN: *pppppp* ASG-ZARA/AUTOMEDIA RETENTION TYPE CATALOG
DFSMSRMM RETENTION TYPE DAYS COUNT(*nnnnnn*) LASTREFERENCED

Problem determination: where

pppppp Data set name pattern used.

nnnnnn Number of days.

Severity: Information

Explanation: This message shows the data set name pattern is used to create a VRS record.

System action: None.

User response: None.

CZA3057I DATASET NAME PATTERN: *pppppp* ASG-ZARA/AUTOMEDIA RETENTION TYPE USER
DFSMSRMM RETENTION TYPE DAYS COUNT(*999999*)

Problem determination: where

pppppp Data set name pattern used.

Severity: Information

Explanation: This message shows the data set name pattern is used to create a VRS record.

System action: None.

User response: None.

CZA3058I DATASET NAME PATTERN: *pppppp* ASG-ZARA/AUTOMEDIA RETENTION TYPE CATALOG
DFSMSRMM RETENTION TYPE CYCLES COUNT(*cccccc*)

Problem determination: where

pppppp Data set name pattern used.

cccccc Number of cycles.

Severity: Information

Explanation: This message shows the data set name pattern is used to create a VRS record.

System action: None.

User response: None.

CZA3059I DFSMSRMM RETENTION TYPE NEXTVRS DAYS COUNT(*dddddd*) ADDED

Problem determination: where

dddddd Number of days.

Severity: Information

Explanation: This message shows the data set name pattern is used to create a VRS record.

System action: None.

User response: None.

CZA3060I DATASET NAME PATTERN: *pppppp* ASG-ZARA/AUTOMEDIA RETENTION TYPE DAYS(*rr*)
DFSMSRMM RETENTION TYPE DAYS COUNT(*dddddd*)

Problem determination: where

pppppp Data set name pattern used.

rr ASG-ZARA/AUTOMEDIA retention type one of RT or RN.

dddddd Number of days.

Severity: Information

Explanation: This message shows the data set name pattern is used to create a VRS record.

System action: None.

User response: None.

CZA3061I DATASET NAME PATTERN: *pppppp* ASG-ZARA/AUTOMEDIA RETENTION TYPE DAYS
DFSMSRMM RETENTION TYPE DAYS COUNT(*cccccc*)

Problem determination: where

pppppp Data set name pattern used.

cccccc Number of days.

Severity: Information

Explanation: This message shows the data set name pattern is used to create a VRS record.

System action: None.

User response: None.

CZA3062I DATASET NAME PATTERN: *pppppp* ASG-ZARA/AUTOMEDIA RETENTION TYPE DAYS +
CATALOG DFSMSRMM RETENTION TYPE DAYS COUNT(*dddddd*)

Problem determination: where

pppppp Data set name pattern used.

dddddd Number of days.

Severity: Information

Explanation: This message shows the data set name pattern is used to create a VRS record.

System action: None.

User response: None.

CZA3063I DATASET NAME PATTERN: *pppppp* ASG-ZARA/AUTOMEDIA RETENTION TYPE DAYS +
CATALOG DFSMSRMM RETENTION TYPE DAYS COUNT(*99999*) WHILECATALOGED
DFSMSRMM RETENTION TYPE NEXTVRS DAYS COUNT(*dddddd*) EXTRADAYS ADDED

Problem determination: where

pppppp Data set name pattern used.

dddddd Number of days.

Severity: Information

Explanation: This message shows the data set name pattern is used to create a VRS record.

System action: None.

User response: None.

CZA3100E RECORD NOT PROCESSED: *rrrrrr*

Problem determination: where

rrrrrr Shows the current input record.

Severity: Error

Explanation: The current input record does not match any specified record type.

System action: Process ends with return code = 12.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA310I INACTIVE OSCD *nnn* IS NOT PROCESSED: *rrrrrr*

Problem determination: where

nnn Shows the current off-site candidate data record number.

rrrrrr Shows the current off-site candidate data record.

Severity: Error

Explanation: The currently processed off-site candidate data record is marked as 'inactive'.

System action: The VRS record creation is skipped.

User response: None.

CZA3200I EDCD PROCESSED: *mmmmmm* *jjjjj*

Problem determination: where

mmmmmm
Current processed data set mask.

jjjjj Current processed job mask.

Severity: Information

Explanation: None.

System action: None.

User response: None.

CZA3201I CHANGED TO: *mmmmmm* *jjjjj*

Problem determination: where

mmmmmm
Current processed data set mask.

jjjjj Current processed job mask.

Severity: Information

Explanation: The data set mask and job mask after the changes to fit DFSMSrmm vital record specification rules.

System action: Change the ASG-ZARA/AutoMedia pattern to DFSMSrmm masking.

User response: None.

CZA3202I OLD DSN MASK *dddddd* CHANGED

Problem determination: where

dddddd Current processed data set mask.

Severity: Information

Explanation: The data set mask before any changes was done.

System action: None.

User response: None.

CZA3203E DSN LEVEL 1 dddddd HAS MORE THAN 8 CHARS

Problem determination: where

dddddd Current processed data set mask.

Severity: Error

Explanation: The current processed data set mask has more than 8 characters in the first level qualifier.

System action: Process ends with return code = 12.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA3204E DSN LEVEL 1 dddddd HAS AN INVALID CHAR

Problem determination: where

dddddd Current processed data set mask.

Severity: Error

Explanation: The current processed data set mask has an invalid character in the first level qualifier. The first level qualifier consists of one-to-eight alphanumeric characters, \$,# or @. The first character cannot be a number.

System action: Process ends with return code = 12.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA3205E DSN LEVEL 1 dddddd HAS AN INVALID FIRST CHAR

Problem determination: where

dddddd Current processed data set mask.

Severity: Error

Explanation: The current processed data set mask has an invalid first character in the first level qualifier. The first level qualifier consists of one-to-eight alphanumeric characters, \$,# or @. The first character cannot be a number.

System action: Process ends with return code = 12.

User response: Check the ASG-ZARA/AutoMedia input records, correct the error and rerun the conversion process.

CZA3206I OLD DSN MASK dddddd CHANGED NEW DSN MASK nnnnnn

Problem determination: where

dddddd Current processed data set mask.

nnnnnn New data set mask.

Severity: Information

Explanation: The data set mask before any changes were done.

System action: The data set mask is changed by the process.

User response: None.

CZA3207I OLD DSN MASK dddddd CHANGED NEW DSN MASK nnnnnn

Problem determination: where

dddddd Current processed data set mask.

nnnnnn New data set mask.

Severity: Information

Explanation: The data set mask before any changes were done.

System action: The data set mask is changed by the process.

User response: None.

CZA3208I OLD DSN MASK *dddddd* CHANGED NEW DSN MASK *nnnnnn* ADDED

Problem determination: where

dddddd Current processed data set mask.

nnnnnn New data set mask.

Severity: Information

Explanation: The data set mask before any changes was done.

System action: The data set mask is changed by the process.

User response: None.

CZA3209I CHANGED TO *nnnnmmm* *jjjjj*

Problem determination: where

nnnnmmm
Current processed data set mask.

jjjjj Current processed job mask.

Severity: Information

Explanation: The data set mask and job mask after the changes to fit DFSMSrmm vital record specification rules.

System action: Change the ASG-ZARA/AutoMedia pattern to DFSMSrmm masking.

User response: None.

CZA3210I OLD DSN MASK *dddddd* CHANGED NEW DSN MASK *nnnnnn*

Problem determination: where

dddddd Current processed data set mask.

nnnnnn New data set mask.

Severity: Information

Explanation: The data set mask before any changes was done.

System action: The data set mask is changed by the process.

User response: None.

CZA3220E DUPLICATE AUTO EXPIRATION DATE CANDIDATE DATA RECORD FOUND: *rrrrrr*

Problem determination: where

rrrrrr Shows the double auto expiration date candidate data record.

Severity: Error

Explanation: The duplicate records are found.

System action: Only first record will be processed.

User response: None.

CZA3221E DUPLICATE AUTO OFF-SITE CANDIDATE DATA RECORD FOUND: *rrrrrr*

Problem determination: where

rrrrrr Shows the double auto off-site candidate data record.

Severity: Error

Explanation: The duplicate records are found.
System action: Only first record will be processed.
User response: None.

CZA3301I SECONDARY VAULT EXPIRATION *rrrrrr* IGNORED

Problem determination: where
rrrrrr Shows the current vault expiration.
Severity: Information
Explanation: Secondary vault expiration record is found.
System action: Record is skipped.
User response: None.

CZA3304E THIRD OR HIGHER VAULT EXPIRATION FOUND FOR *dddddd* LOC: *lllll* (*nnn*)

Problem determination: where
dddddd Data set name.
lllll Location name.
nnn Vault expiration number.
Severity: Error
Explanation: Third or higher vault expiration record is found.
System action: Record is skipped.
User response: None.

CZA3305E UNKNOWN EXPIRATION FOUND FOR *dddddd*

Problem determination: where
dddddd Data set name.
Severity: Error
Explanation: Unknown expiration type is found.
System action: Record is skipped.
User response: None.

CZA3310I VAULT RETENTION FOR DSN *dddddd* COUNT *nnnnn* *ttttt*

Problem determination: where
dddddd Data set name.
nnnnn Retention amount.
ttttt Retention type.
Severity: Information
Explanation: Shows the retention criteria which is used to create a VRS record.
System action: None.
User response: None.

CZA3700I OSCD PROCESSED: *dddddd jjjjj*

Problem determination: where

dddddd Data set name.

jjjjj Job.

Severity: Information

Explanation: Shows the off-site candidate data record which is processed.

System action: None.

User response: None.

CZA3701I EXPIRATION AND VAULT RETENTION MATCHED FOR DATA SET MASK: *dddddd jjjjj* EDCD LOCATION: *lllll1* OSCD LOCATION: *lllll2* DIFF. LOCATIONS: *nnn*

Problem determination: where

dddddd Data set name.

jjjjj Job.

lllll1 Expiration date candidate location.

lllll2 Off-site candidate location.

nnn Different locations count.

Severity: Information

Explanation: Exact match of retention and vault definitions.

System action: None.

User response: None.

CZA3702W EXPIRATION AND VAULT RETENTION MATCHED BY USING A GENERIC MASK GENERIC VAULT MASK: *dddddd_jjjjj1* EXP. DATA SET MASK: *dddddd_jjjjj2* EDCD LOCATION: *lllll1* OSCD LOCATION: *lllll2* DIFF. LOCATIONS: *nnn*

Problem determination: where

dddddd_jjjjj1
Generic vault mask.

dddddd_jjjjj2
Expiration date candidate data set mask.

lllll1 Expiration date candidate location.

lllll2 Off-site candidate location.

nnn Different locations count.

Severity: Warning

Explanation: Exact match of retention and vault definitions.

System action: None.

User response: None.

CZA3703I EXPIRATION AND VAULT RETENTION MATCHED BY USING A GENERIC VAULT MASK DATA SET MASK: *dddddd_jjjjj* GENERIC MASK: *ddddddd* USED EDCD LOCATION: *lllll1* OSCD LOCATION: *lllll2* DIFF. LOCATIONS: *nnn*

Problem determination: where

dddddd Data set name.

jjjjj Job.

ddddddd

Generic mask.

lllll1 Expiration date candidate location.

lllll2 Off-site candidate location.

nnn Different locations count.

Severity: Information

Explanation: Exact match of retention and vault definitions.

System action: None.

User response: None.

CZA3705I VAULT VRS DEFINITION CREATED FOR DATA SET MASK: *dddddd jjjjj*

Problem determination: where

dddddd Data set name.

jjjjj Job.

Severity: Information

Explanation: Vault VRS definition created.

System action: None.

User response: None.

CZA3710I VRS RECORD CREATION FOR: *dddddd lllll*

Problem determination: where

dddddd Data set name.

lllll Location.

Severity: Information

Explanation: The VRS record is created during migration of retention and movement information.

System action: None.

User response: None.

CZA3222W AUTO OFF-SITE CANDIDATE DATA RECORD HAS NO EXACT CORRESPONDING AUTO EXPIRATION DATE CANDIDATE DATA RECORD: DSN= *dddddd* JOB= *jjjjj*

Problem determination: where

dddddd Data set name.

jjjjj Job.

Severity: Warning

Explanation: No matching retention definition found.

System action: None.

User response: None.

CZA3223I USED MASK: DSN= *dddddd* JOB= *jjjjj*

Problem determination: where

dddddd Data set name.

jjjjj Job.

Severity: Information

Explanation: The specified data set mask has been used to check a matching expiration definition for off-site definitions.

System action: None.

User response: None.

CZA3224E NO MATCHING GENERIC AUTO EXPIRATION DATE CANDIDATE DATA RECORD FOUND

Severity: Error

Explanation: No matching retention definition found for off-site definition.

System action: None.

User response: None.

CZA3800W ALL GENERIC NAMES IN POOL XD*nnnnn* ARE IN USE

Problem determination: where

nnnnn VRS number.

Severity: Warning

Explanation: Next VRS name can not be generated since there is no empty names in XD*nnnnn* pool.

System action: VRS name XD*nnnnn* is generated without generic alphabetic suffix.

User response: Manually change XD*nnnnn* to appropriate VRS name.

Am i ready to continue?

In this chapter we have covered all of the activities that you should have considered before converting from ASG-ZARA/AutoMedia to DFSMSrmm.

After completing the activities in this chapter, you should have the following valid and tested input to load the DFSMSrmm CDS:

- Input to exits.
- Correct PARMLIB options.
- Housekeeping jobs to use in an DFSMSrmm environment.
- Changed reports.
- Extract and convert control statements.
- Extracted data from the ASG-ZARA/AutoMedia DBD.

If the data extract program has error return codes, read the error messages to determine the actions required to correct the errors. Then rerun the extraction process.

To have a valid DFSMSrmm CDS, you must stop all tape activity when you get the data from the ASG-ZARA/AutoMedia database until you start DFSMSrmm in record-only mode in parallel with ASG-ZARA/AutoMedia.

Before producing a valid conversion, you can use a copy of the ASG-ZARA/AutoMedia database to test the extraction and conversion process, so you do not have to stop the tape activities every time you find that you have to go back and rerun the extract jobs.

Once you have tested the environment, you should have the correct tailored input to the extract program.

You can now stop tape activities and produce a valid data extraction from the ASG-ZARA/AutoMedia DBD, RDS, and RMF.

If your previous tests were satisfactory, you should obtain the correct input to the EDGCNVT program and proceed to the next step.

Chapter 9. Building the DFSMSrmm CDS

In this chapter we describe the DFSMSrmm EDGCNVT program and how to use it to build an DFSMSrmm CDS. The input to the EDGCNVT program consists of records that describe the volumes, data sets, and policies you want to convert to DFSMSrmm. In addition, you can control EDGCNVT program processing by using SYSIN control statements that can edit the input records.

The output of the EDGCNVT program includes:

- Records ready for the DFSMSrmm CDS
- DFSMSrmm ADDVRS subcommands that interpret the processed policies
- Control statements for use with the EDGUTIL utility
- Messages that describe processing problems and list statistics.

Objectives of chapter

- Explain how to use the EDGCNVT conversion program
- Convert extracted data
- Build a ready-to-use DFSMSrmm CDS containing converted data.

Audience

- System programmers
- Storage administrators.

At the end of this chapter you will have built an DFSMSrmm CDS ready for use in either a testing, parallel running, or cutover to production environment.

Preparing to run EDGCNVT

Once you have the extracted data from the vendor database or any other data source you must use the EDGCNVT program to convert the data to a format usable in the DFSMSrmm CDS.

Before running EDGCNVT, check your environment for:

- Migration of system-managed tape volumes
- Different pool prefixes
- The best conversion method for VRS policies.
- In some cases the running of EDGCNVT requires that DFSMSrmm is first started in manual mode, or any other mode, so that values from the DFSMSrmm parmlib are available for use by EDGCNVT. See below.

After running EDGCNVT, optionally run the EDGCEBIN program.

If you have vendor specific, non-IBM, media then DFSMSrmm should be started in the manual mode before running EDGCNVT. Vendor specific media information should be specified on the MEDINF command in the DFSMSrmm PARMLIB member. A sample MEDINF table is shipped as EDGMEDST member of SAMPLIB.

System-managed Volumes

The EDGCNVT input records do not contain information to indicate whether a location is a system-managed library or a volume is system-managed. At the time you extracted data from your existing tape management system you may have supplied a location name for every volume belonging to

an automated tape library. However, you probably do not have a specific location name for system-managed libraries in your previous tape management system.

You must supply the specific location information by using EDGCNVT control statements. There are three options:

- LOCDEF

Use this statement to define a location as being system-managed and whether it is an IBM 3494 or IBM 3495 Automated Tape Library (ATL) or a IBM 3495-M10 Manual Tape Library (MTL)

Note: NOTE: Any customer that has System Managed Tape and using LOCDEF statements for EDGCNVT SYSIN DD should plan to run EDGUTIL with VERIFY(VOLCAT) to ensure that the RMM CDS built by conversion is in synch with the TCDB. They could combine the EDGUTIL VERIFY(ALL) with the VOLCAT, by using EDGUTIL with VERIFY(VOLCAT,ALL). See “Verifying the DFSMSrmm CDS” on page 220 for more details on how to verify the RMM CDS.

- IF VOLRANGE

Use this statement to identify the volumes that are system-managed. If you know the range of volumes that are inside the system-managed library or have the system-managed library as a home location, you can assign a specific location name to this range. You only need to do this if the input records do not already have the correct location and home location names.

- IF STORLOC

If you have a specific location name for the volumes belonging to the library and it is different from the name to be used under DFSMSrmm, you can translate it to the location name that you plan to use in the DFSMSrmm environment.

For an explanation of these statements, see “Running EDGCNVT and EDGCEBIN” on page 187

Remember that in a system-managed DFSMSrmm environment the location name you specify for the tape library must match the name of the library defined to SMS.

Pool Prefixes

If you plan to move your volumes using media name criteria, you must ensure that you have correctly supplied media names in the extraction process by using the VOLMEDIA statement for TLMS migration and the hard coded values for CA-1.

You can optionally use the IF VMEDIA control statement of the EDGCNVT program to translate the media names in the input records to the values you plan to use with DFSMSrmm. For an explanation of these statements see “Running EDGCNVT and EDGCEBIN” on page 187. Ensure that the media names you use match the values specified on the DFSMSrmm PARMLIB LOCDEF and VLPOOL options.

VRS Conversion

When you run EDGCNVT specifying the VRSCMDS DD-statement, all ADDVRS commands are built from the input K-Records to match the retention and movement criteria. If you are not happy with the VRS records built by the combination of extract programs and EDGCNVT program, you can edit the commands. This could be an ideal opportunity to simplify your retention and movement policies to exploit the DFSMSrmm generic data set name mask capabilities.

If you edit the commands, you can execute the command stream once you have loaded the remaining data created by EDGCNVT into an DFSMSrmm CDS. Remember to remove the VRSLIST file from the EDGJLOAD job so that the VRS records created by EDGCNVT do not end up in the DFSMSrmm CDS.

See “Adding VRSs” on page 221 for additional information on the VRS conversion process.

Movement Tracking Date

The data set primary VRS subchain start date for all data sets on a volume is set from the volume movement tracking date. The movement tracking date is currently set from the CA-1 outcode date (date assigned to a vault). EDGHSKP VRSEL processing uses this date and the volume current location to determine how and if, to process EXTRADAYS VRSs on the first VRSEL run after conversion.

EDGCNVT Considerations for Expiration Date

The expiration dates of volumes and data sets on **Retention Method EXPDT** managed volumes are processed at conversion time as follows:

- **The data set expiration date** of a single data set or of the first file of a multi volume data set is set to the maximum of the data set expiration date and the date calculated using data set LASTREF extra_days. The LASTREF extra_days is added to the date of last reference or in case the date of last reference is zero added to the creation date. All subsequent files of a multi volume data set inherit the expiration date of the first file of a multi volume data set. For information how DFSMSrmm re-determines the expiration date during the life of a data set or volume please refer to the DFSMSrmm Implementation & Customization Guide, Chapter 3 Retention Method.
- **The volume expiration date** is processed depending on the volume attribute RETAINBY. For volumes that are retained by VOLUME or SET the expiration date is set to the highest date of the volume expiration date and the expiration dates of the data sets residing on the volume. Additionally for volumes retained by SET the highest volume expiration date of the volume set is propagated to all volumes in the set. For volumes retained by FIRSTFILE the expiration date of the first file on a single volume or on the first volume in a multi volume set is propagated to the single volume and respectively to all volumes in the multi volume set. In case the volume expiration date copied from the first file is lower than the volume expiration date the warning message EDGCNVT-00180 215 is issued.

The expiration dates of volumes and data sets on **Retention Method VRSEL** managed volumes are converted with no additional processing, except all subsequent files of a multi volume data set inherit the data set expiration date of the first file of a multi volume data set.

Running EDGCNVT and EDGCEBIN

EDGCNVT is a batch program that takes the records from the input data set, reformats them, and makes them ready for the DFSMSrmm CDS. The input records are usually the records you created by running one of the sample extract processes, but you can construct the records using any technique or tool that you prefer.

As each record is read, it is passed to the EDGCNVT user exit, if you have supplied one, where it can optionally be updated or skipped. Upon return from the user exit, EDGCNVT compares each record encountered for applicability to the control statements in the SYSIN data set and modifies the records accordingly. It then generates the output records and data sets, which are ready for postprocessing and ultimately entered into the DFSMSrmm CDS.

The SYSIN data set record format can be found in “SYSIN Control Statements” on page 191.

EDGCEBIN is a batch program that appends the BINLIST output file, created by EDGCNVT, with new empty bin records.

The following figure shows the sample JCL for execution of EDGCNVT and EDGCEBIN

We describe the input and output files in the sections that follow.

```

//CONVERT JOB ....
//*
//STEP1 EXEC PGM=EDGCNVT,REGION=6M
//*STEPLIB DD DISP=SHR,DSN=STSGWD.RMM.LINKLIB
//SYSPRINT DD SYSOUT=*
//SYSOUT DD DISP=(,CATLG),LRECL=80,RECFM=FB,AVGREC=K,
//          SPACE=(80,(1,1),RLSE),UNIT=SYSALLDA,
//          DSN=STSGWD.EDGCNVT.SYSOUT.DATA
//VRSCMDS DD DISP=(,CATLG),LRECL=80,RECFM=FB,AVGREC=K,
//          SPACE=(80,(1,1),RLSE),UNIT=SYSALLDA,
//          DSN=STSGWD.EDGCNVT.VRSCMDS.DATA
//DEXTRCT DD DISP=SHR,DSN=STSGWD.EDGCXLDL.DEXTOUT.DATA
//          DD DISP=SHR,DSN=STSGWD.EDGCSRDS.DEXTOUT.DATA
//          DD DISP=SHR,DSN=STSGWD.EDGCSVDS.DEXTOUT.DATA
//          DD DISP=SHR,DSN=STSGWD.EDGCXBIN.DEXTOUT.DATA
//LIBLIST DD DISP=(,CATLG),LRECL=1000,RECFM=VB,AVGREC=K,
//          SPACE=(1000,(70,10),RLSE),UNIT=SYSALLDA,
//          DSN=STSGWD.EDGCNVT.LIBLIST.DATA
//OWNLIST DD DISP=(,CATLG),LRECL=400,RECFM=VB,AVGREC=K,
//          SPACE=(400,(1,1),RLSE),UNIT=SYSALLDA,
//          DSN=STSGWD.EDGCNVT.OWNLIST.DATA
//BINLIST DD DISP=(,CATLG),LRECL=256,RECFM=VB,AVGREC=K,
//          SPACE=(256,(20,10),RLSE),UNIT=SYSALLDA,
//          DSN=STSGWD.EDGCNVT.BINLIST.DATA
//VRSLIST DD DISP=(,CATLG),LRECL=256,RECFM=VB,AVGREC=K,
//          SPACE=(256,(20,10),RLSE),UNIT=SYSALLDA,
//          DSN=STSGWD.EDGCNVT.VRSLIST.DATA
//SYSIN DD *
RMMCD5 EQUALS MYCDSID
IF OWNER EQUALS ' ' THEN DEFOWNR EQUALS 'USER001 '
LOCDEF MTLNAME MTL
LOCDEF ATLNAME ATL
IF VOLRANGE IS 200000 TO 205000 THEN SET LOCATION ATLNAME
IF VMEDIA EQUALS CART3480 THEN MEDIANAME EQUALS CARTS
IF VMEDIA EQUALS CART3490 THEN MEDIANAME EQUALS CARTS
IF STORLOC EQUALS LIB THEN RMMSTORE EQUALS SHELF
IF STORLOC EQUALS LIBR THEN RMMSTORE EQUALS SHELF
/*

```

Figure 49. Sample EDGCNVT and EDGCEBIN Execution JCL (Part 1 of 2)

```

//*****
//*
//*   SORT BINLIST BY User-Store Name, User-Store Bin Media Name,
//*                   User-Store Bin Name
//*
//*****
//SORT      EXEC PGM=SORT,REGION=4M
//SYSOUT    DD SYSOUT=*
//SORTIN    DD DISP=SHR,DSN=STSGWD.EDGCNVT.BINLIST.DATA
//SORTOUT   DD DISP=(,PASS),
//           DSN=STSGWD.EDGCNVT.BINLIST.DATA.SORT,
//           RECFM=VB,LRECL=256,AVGREC=K,UNIT=SYSDA,
//           SPACE=(256,(20,10),RLSE)
//SYSIN     DD *
SORT FIELDS=(7,8,CH,A,15,8,CH,A,23,6,CH,A),DYNALLOC
INCLUDE COND=((5,1,CH,EQ,C'S'),OR,      Include only Bin Records
              (5,1,CH,EQ,C'R'))        (Empty and Assigned)
              OPTION VLSHRT,NOCHALT    Avoid msg ICE218A
//*****
//*
//*   RUN EDGCEBIN TO FILL THE GAPS BETWEEN USED BINS
//*                   WITH EMPTY BINS IN THE BINLIST FILE.
//*
//*   NOTE: SPECIFY THE PREFIX LENGTH VALUE VIA PARM='PREFLEN(1)'
//*         PREFLEN(0) IS DEFAULT
//*****
//CEBIN     EXEC PGM=EDGCEBIN,PARM='PREFLEN(2)'
//SORTBIN   DD DISP=(OLD,DELETE),UNIT=SYSALLDA,
//           DSN=STSGWD.EDGCNVT.BINLIST.DATA.SORT
//BINLIST   DD DISP=(MOD,KEEP),UNIT=SYSALLDA,
//           DSN=STSGWD.EDGCNVT.BINLIST.DATA
//SYSPRINT  DD SYSOUT=*
//

```

Figure 49. Sample EDGCNVT and EDGCEBIN Execution JCL (Part 2 of 2)

The sample JCL to run the EDGCNVT and EDGCEBIN programs is provided in SAMPLIB member EDGJCNVT.

You should modify the execution parameter values, data set names, and data set attributes and specify the SYSIN control statements you need before using the sample job.

Execution Parameters

EDGCNVT has a single, optional execution parameter.

Table 75. EDGCNVT Execution Parameters

Parameter	Description
nmmn	A number of one or more digits that specifies the maximum number of LOCDEF, VOLRANGE or VOLPRE statements to support. The default value is 500 of each type. Specify a value if you want to increase the limit.

EDGCEBIN has a single, optional execution parameter.

Table 76. EDGCEBIN Execution Parameters

Parameter	Description
PREFLEN(l)	l is a number of one digit, from 0 to 5, that specifies the prefix length of storage bin numbers in the DFSMSrmm storage location. $l = 6 - n$, where n is the length of the bin numbers used in the EDGCEBIN program. For example, if you convert from TLMS to DFSMSrmm and want to consider the TLMS storage location id as a prefix and cabinet and box numbers as a bin number, specify PREFLEN(2). The default value is PREFLEN(0).

EDGCNVT Input Files

Following table lists the input files used by EDGCNVT.

Table 77. EDGCNVT Input Files

DDNAME	Description
DEXTRCT	This file contains the records that are to be converted to DFSMSrmm format. See "DEXTRCT" for information about the input records.
SYSIN	The SYSIN control statements describe the processing that you would like EDGCNVT to perform on the records to be converted. See "SYSIN Control Statements" on page 191 for information about the control statements.

DEXTRCT

The DEXTRCT file contains the records that are input to EDGCNVT for conversion to DFSMSrmm record format. You can either use the DFSMSrmm-supplied sample extract programs to create the input records or generate them using any tool or method you prefer. This record-based interface provides you with a clearly defined way to convert volume tape resource data into a format supported by DFSMSrmm.

There are five different records that describe input to EDGCNVT:

Record Type

Description

- L** Library. An L-Record contains details of a single volume and the first data set on that volume.
- D** Data set name. A D-Record contains details of a single data set. One D-Record is required for each additional data set on a volume. If you supply a D-Record for a file, all previous files on the same volume must be specified in D-Records that precede the record.

The L- and D-Records have the following sequence requirements:

- Library (L) records for a multi volume set of volumes must appear in ascending volume sequence order, interrupted only by:
Data set name (D) records, in ascending file sequence order, after each L-Record that is created for the volumes from which the data set originates.
- The count of D-Records that follow after a nonscratch L-record must be one less than the count of data sets in the LVDSNCNT field of that L-Record because each L-Record contains first file data set information.

Note: D-Records are not supported for scratch volumes.

- E** Empty rack and bin. The E-Record contains details of a single, empty shelf location. It can describe either a rack number or a bin number.
- O** Owner. The O-Record contains details of a single owner ID. The owner may own resources that are being converted, but it need not own any resources.
- K** VRS. The K-Record contains details of a single policy definition.

The EDGCNVT program input records have the following content requirements:

- The intention is that you use the DFSMSrmm installation-wide exit, EDGUX100, to check the JCL-specified EXPDT value. You should also use that exit to check for a special date of 98000 (other vendor product's "ignore volume"). When the date is found, the exit requests, through the parameter list, that DFSMSrmm ignore the volume. The implication for migration, and therefore for extract processing, is that no volumes with identical VOLSERS can be input to EDGCNVT.
- Data sets that are retained by special keyword dates in other tape management systems and are to be associated with a special expiration date VRS Management value must either:
 - Contain the VRS Management value in either the D-Record DSVRSVAL field or the L-Record LDVRSVAL field in the extract program output data set
 - or
 - Have the VRS management value satisfied through the user exit (see "EDGCNVT User Exit" on page 204) by updating the LDVRSVAL field in an L-Record or the DSVRSVAL in a D-Record.
- The security level of data sets and volumes should be specified through either the appropriate fields in the L-Records and D-Records or a series of policy and rules statements in SYSIN. The values for security level should be the same as those established through PARMLIB.

SYSIN Control Statements

EDGCNVT accepts a series of statements from SYSIN in the format shown in following Figure.

```
IF fieldname1 EQUALS fieldname2 THEN fieldnamex EQUALS fieldnamey
                        or
                        valuexxxxx                          or
                        valueyyyyy
```

Figure 50. Convert Program VRS Input Record Format

EDGCNVT RECORD CREATE DATE, TIME, CDS AND SYSTEM IDENTIFICATION: EDGCNVT accepts the control statements shown in the following figure. If the RMMCDS statement is not included, EDGCNVT processing runs as if RMMCDS EQUALS "CDSID" was specified. If the other statements are not included, EDGCNVT acquires the values from the operating system at the time of each execution. It uses these values to date, time stamp, and system identify the output records.

```
RMMCDS EQUALS cdsid
RMMCRD EQUALS yyyyddd
RMMCRT EQUALS hhmsst
RMMCRS EQUALS systemid
```

Figure 51. EDGCNVT Record Create Date, Time, CDS and System Identification

YYYYDDD

Is the creation date used on all records converted by EDGCNVT.

HHMMSST

Is the creation time used on all records converted by EDGCNVT. The time is specified as hh - hours, mm - minutes, ss - seconds, and t - tenths of seconds.

SYSTEMID

This is the system identifier used to identify the system on which a record is created. The value you specify should match the value you specify on the DFSMSrmm PARMLIB OPTION SYSID operand.

CDSID

This is the identifier of the control data set that must be used on this system. The value you specify should match the value you specify on the DFSMSrmm PARMLIB OPTION CDSID operand.

VRS RELEASE OPTIONS PROCESSING: Use this statement to force or prevent use of VRS release options EXPIRYDATEIGNORE and/or SCRATCHIMMEDIATE.

```
VRS vo choice
```

Figure 52. VRSREL Release action

VO VRS release option, one of XDI or SI

XDI EXPIRYDATEIGNORE, DFSMSrmm ignores the volume expiration date when no other vital record specification matches to the data set.

SI SCRATCHIMMEDIATE, DFSMSrmm returns the volume to scratch status when the only pending release action for the volume is to return to scratch.

CHOICE

ON or OFF

ON Add specified VRS release option to each created DSN VRS. Override what was set from the previously running conversion tools.

OFF (Default) Never add specified VRS release option to any created DSN VRS. Override what was set from the previously running conversion tools.

Note: Not specifying VRS XDI and/or VRS SI means use the value set from previous conversion tools.

ASSIGNING A DEFAULT RELEASE ACTION: Use this statement to assign a release action to any volume that does not have a release action specified.

```
IF RETACTN EQUALS '00' THEN RETACTN EQUALS SCRPOOL
```

Figure 53. RETACTN Release action

'00' A hexadecimal value that identifies there are no release actions.

SCRPOOL

Requests that 'return to scratch' release action be assigned to all volumes that have no release action.

This is the only IF RETACTN statement that is valid.

TRANSLATING LOCATION NAMES: Use this statement to translate the name of locations in L-, E-, and K-Records in the DEXTRCT file to values that you plan to use with DFSMSrmm. Only do this if the location names are to be different.

If, for example, your current tape management system refers to the tape library by the name TLIBRARY, the statement shown in the following figure will change that name to SHELF, as DFSMSrmm requires. If your current tape management system uses OFFSITE1, OFFSITE2, or OFFSITE3, the statement shown in the following figure will change the names to LOCAL, DISTANT, and REMOTE, respectively.

```

IF STORLOC EQUALS TLIBRARY THEN RMMSTORE EQUALS SHELF
      or
IF STORLOC EQUALS OFFSITE1 THEN RMMSTORE EQUALS LOCAL
      or
IF STORLOC EQUALS OFFSITE2 THEN RMMSTORE EQUALS DISTANT
      or
IF STORLOC EQUALS OFFSITE3 THEN RMMSTORE EQUALS REMOTE
      or
IF STORLOC EQUALS name      THEN RMMSTORE EQUALS location

```

Figure 54. IF STORLOC SYSIN Statement

Note: STORLOC and RMMSTORE are in the records produced by input to EDGCNVT.

The normal way to use the IF STORLOC statement is together with a LOCDEF statement. The LOCDEF defines the location name and location type as used with DFSMSrmm. If you do not define locations other than SHELF, LOCAL, DISTANT, and REMOTE with LOCDEF, you cannot use the IF STORLOC statement.

Note: Any customer that has System Managed Tape and using LOCDEF statements for EDGCNVT SYSIN DD should plan to run EDGUTIL with VERIFY(VOLCAT) to ensure that the RMM CDS built by conversion is in synch with the TCDB. They could combine the EDGUTIL VERIFY(ALL) with the VOLCAT, by using EDGUTIL with VERIFY(VOLCAT,ALL).

If your current tape management system refers to VLT1 and you want to use the location name VAULT1 with DFSMSrmm, use the statements shown in the following figure.

```

LOCDEF VAULT1  BINS
IF STORLOC EQUALS VLT1      THEN RMMSTORE EQUALS VAULT1

```

Figure 55. IF STORLOC Example with LOCDEF

If your current tape management system refers to the inhouse disaster storage area for disaster recovery tapes as ONSITE, the statement shown in following figure will change that name to LOCAL.

```

IF STORLOC EQUALS ONSITE  THEN RMMSTORE EQUALS LOCAL

```

Figure 56. IF STORLOC Onsite Example

If your current tape management system refers to the in-town disaster storage area for disaster recovery tapes as OFFSITE, the statement shown in the following figure will change that name to DISTANT.

```

IF STORLOC EQUALS OFFSITE THEN RMMSTORE EQUALS DISTANT

```

Figure 57. IF STORLOC Offsite Example

If your current tape management system refers to the out-of-town disaster storage area for disaster recovery tapes as IRONMTN, the statement shown in the following figure will change that name to REMOTE.

```

IF STORLOC EQUALS IRONMTN THEN RMMSTORE EQUALS REMOTE

```

Figure 58. IF STORLOC Remote Example

If you need embedded blanks in the location name specify STORLOC value in quotes, use the statements shown in the following figure as a sample.

```
LOCDEF VAULT2 BINS
IF STORLOC EQUALS 'VL T2' THEN RMMSTORE EQUALS VAULT2
IF STORLOC EQUALS 'ON SITE' THEN RMMSTORE EQUALS LOCAL
```

Figure 59. IF STORLOC with Blank Example

SPECIFYING SECURITY LEVELS: If you would like to establish a security level for data sets that meet a specific naming convention, use the statement shown in the following figure as a sample.

```
IF DSN EQUALS SYS1.IC.IC.** THEN SECLEVEL EQUALS 11
```

Figure 60. IF DSN SYSIN Statement Syntax

Note: Only asterisks (*) are supported as filters, and they must be last in the data set name or VOLSER.

The security level number assigned must be a one- to three-digit number and have a value between 0 and 255 as defined by the DFSMSrmm SECLEV option in PARMLIB.

If you would like to establish a security level for volumes that meet a specific naming convention, use the statement shown in the following figure as a sample.

```
IF VOL EQUALS SY** THEN SECLEVEL EQUALS 009
```

Figure 61. IF VOL SYSIN Statement Syntax

SPECIFYING AND TRANSLATING OWNER INFORMATION: If, as the EDGCNVT program is executing, you would like to have owner information added to those volumes and data sets that currently do not have owner information use the example shown in the following figure with your default values.

```
IF OWNER EQUALS '      ' THEN DEFOWNR EQUALS 'library '
IF OWNER EQUALS '      ' THEN DEFOWNS EQUALS 'sur name '
IF OWNER EQUALS '      ' THEN DEFOWNF EQUALS '1st name '
IF OWNER EQUALS '      ' THEN DEFOWND EQUALS 'dept num '
IF OWNER EQUALS '      ' THEN DEFOWNT EQUALS 'int tele'
IF OWNER EQUALS '      ' THEN DEFOWNU EQUALS 'userid '
IF OWNER EQUALS '      ' THEN DEFOWNN EQUALS 'nodeid '
```

Figure 62. IF OWNER SYSIN Statement Syntax for Default Owner

LIBRARY

One- to eight-character owner ID to be used as a default value

SUR NAME

One- to 20-character surname or family name for the default owner ID

1ST NAME

One- to 20-character first name for the default owner ID

DEPT NUM

One- to 30-character department name for the default owner ID

Note that in Figure 62 the space between the beginning and ending apostrophe is less than 30 characters.

INT TELE

One- to eight-character internal telephone number for the default owner ID

USERID

One- to eight-character userid for the default owner ID

NODEID

One- to eight-character node name for the default owner ID

If, when the EDGCNVT program is executing, you would like to translate one existing owner to a new value, use the example shown in following figure with your new values.

```

IF OWNER EQUALS 'ownerid ' THEN NEWOWNR EQUALS 'newown '
IF OWNER EQUALS 'ownerid ' THEN NEWOWNS EQUALS 'sur name '
IF OWNER EQUALS 'ownerid ' THEN NEWOWNF EQUALS '1st name '
IF OWNER EQUALS 'ownerid ' THEN NEWOWND EQUALS 'dept num '
IF OWNER EQUALS 'ownerid ' THEN NEWOWNT EQUALS 'int tele'
IF OWNER EQUALS 'ownerid ' THEN NEWOWNU EQUALS 'userid '
IF OWNER EQUALS 'ownerid ' THEN NEWOWNN EQUALS 'nodeid '

```

Figure 63. IF OWNER SYSIN Statement Syntax for Translating Owner

OWNERID

One- to eight- character owner ID in the input records to be translated

NEWOWN

One- to eight-character owner ID to be used instead of ownerid

SUR NAME

One- to 20-character surname or family name for the owner ID

1ST NAME

One- to 20-character first name for the owner ID

DEPT NUM

One- to 30-character department name for the owner ID

Note that in Figure 63 the space between the beginning and ending apostrophe is less than 30 characters.

INT TELE

One- to eight-character internal telephone number for the owner ID

USERID

One- to eight-character user ID for the owner ID.

NODEID

One- to eight-character node name for the owner ID

IDENTIFYING LOCATION NAMES AND TYPES: Use the LOCDEF statement in following figure to define locations other than the DFSMSrmm built-in storage locations (LOCAL, DISTANT, REMOTE, or SHELF) to the conversion program.

```

LOCDEF loc_name type home

```

Figure 64. LOCDEF Statement

LOC_NAME

One- to eight-character location name that will be known to DFSMSrmm after conversion.

If the location names in the extract records that are input to EDGCNVT do not match these values, use the IF STORLOC statement to make EDGCNVT perform the translation.

TYPE

One of LOAN, BINS, NOBINS, MTL, or ATL:

LOAN

Defines the location as a loan location. If the extract records have the location as a storage location, EDGCNVT clears the assigned bin number field and sets the volume location to its home location and marks the volume on loan.

BINS Identifies the location as an installation-defined shelf-managed storage location. The extract records must have set a bin number for the volume.

NOBINS

Identifies the location as an installation-defined non-shelf-managed storage location. If the extract records have set a bin number for the volume, it is ignored.

ATL The location is identified as an automated system-managed library.

MTL The location is identified as a manual system-managed library.

HOME

Optionally specify the keyword HOME to indicate that an installation defined storage location can also be a home location.

When you specify HOME for a storage location, this affects the processing performed for the IF VOLRANGE statement with LOCATION. Any volume in the defined range has its home location set to the name of this storage location.

Up to 500 LOCDEF statements can be defined in SYSIN.

IMPLEMENTING STORAGE LOCATIONS AS HOME LOCATIONS: You can use storage locations as the home locations for volumes to decide how volumes are shelf managed. To implement storage locations as home locations:

1. Specify LOCDEF store BINS HOME in the EDGCNVT SYSIN statements
2. Specify IF VOLRANGE with LOCATION to assign location names based on volume ranges.
3. Ensure that LOCDEF commands in RMM parmlib include TYPE(STORAGE,HOME) only for those storage locations to also be home locations

ASSIGNING MEDIA TYPE USING IF VOLPRE: Use the statement shown in following figure to assign a media type and/or media information to a range of volumes based on the VOLSER prefix and suffix. The media type must be either one of IBM media types or a vendor specific media type defined in the EDGRMMxx member of PARMLIB. When you want to use a non-IBM media type you must specify both a MEDINF value and a media type. The non-IBM media type statement must be preceded by a statement that sets a MEDINF name for the same volume range.

```
IF VOLPRE IS ppppp AND VOLSUF IS ssssss TO eeeee THEN SET CST
IF VOLPRE IS ppppp AND VOLSUF IS ssssss TO eeeee THEN SET ECCST
IF VOLPRE IS ppppp AND VOLSUF IS ssssss TO eeeee THEN SET HPCT
IF VOLPRE IS ppppp AND VOLSUF IS ssssss TO eeeee THEN SET EHPCT
IF VOLPRE IS ppppp AND VOLSUF IS ssssss TO eeeee THEN SET MEDINF STK
IF VOLPRE IS ppppp AND VOLSUF IS ssssss TO eeeee THEN SET STK1R
IF VOLPRE IS ppppp AND VOLSUF IS ssssss TO eeeee THEN SET MEDINF STK
IF VOLPRE IS ppppp AND VOLSUF IS ssssss TO eeeee THEN SET STK2P
```

Figure 65. VOLPRE SYSIN Statement

PPPPP

VOLSER prefix. Specify from one to five characters as a prefix.

SSSSSS

The starting range of a VOLSER suffix. Specify up to six characters; the characters must be right aligned, and the total number of prefix and suffix characters must be six.

EEEEEE

The ending range of a VOLSER suffix. Specify up to six characters; the characters must be right aligned, and the total number of prefix and suffix characters must be six.

Up to 500 VOLPRE statements can be defined in SYSIN.

Note: Using vendor specific media type and media information requires DFSMSrmm to be started in the manual mode before running EDGCNVT. The media type and media information should match the values specified on the MEDINF command in the DFSMSrmm PARMLIB member. See the SAMPLIB member EDGMEDST for the MEDINF commands which should be used for conversion records created by EDGC5LDR.

Following figure shows a sample to set the media type to CST for VOLSERs C00001 to C01000 and ECCST for VOLSERs E00001 to E01000.

```
IF VOLPRE IS C      AND VOLSUF IS 00001 TO 01000 THEN SET CST
IF VOLPRE IS E      AND VOLSUF IS 00001 TO 01000 THEN SET ECCST
```

Figure 66. VOLPRE Example

IF VOLRANGE WITH LOCATION: Use the VOLRANGE with LOCATION statement in following figure to assign a location name to a set of volumes based on the VOLSER prefix. The location must be defined by using the LOCDEF statement, or it must be a built-in location name.

```
IF VOLRANGE IS fvolsr TO lvolsr THEN SET LOCATION loc_name
```

Figure 67. VOLRANGE with LOCATION Statement

FVOLSR

The first VOLSER in the range. Specify a one- to six-character VOLSER.

LVOLSR

The last VOLSER in the range. Specify a one- to six-character VOLSER. You can specify the same value as for fvolsr to have a single volume range.

LOC_NAME

The one- to eight-character location name that will be known to DFSMSrmm after conversion.

Up to 500 VOLRANGE with LOCATION statements can be defined in SYSIN.

IF VOLRANGE WITH PREFIX: Use the IF VOLRANGE with PREFIX statement to change the rack number that is used for a set of volumes based on the VOLSER. You can change as many characters in the rack number as you want, based on the volume VOLSER.

For example, you have volumes in the 013500–013999 range in a scratch pool, but you also have a scratch pool with a range of 013000-013499. The RMM VLPOOL prefix of 013* does not provide for two separate pools. However, using VOLRANGE with PREFIX, you can change the rack numbers so that the VLPOOL prefixes are different and get RMM to manage two pools:

```
IF VOLRANGE IS 013000 TO 013496 THEN SET PREFIX P
```

IF VOLRANGE IS 013497 TO 013999 THEN SET PREFIX S

You can still use the significant parts of the VOLSERs to identify volumes, and the rack numbers allow the scratch pool to be identified as well as the VOLSER.

```
IF VOLRANGE IS fvolser TO lvolser THEN SET PREFIX r_pref
```

Figure 68. VOLRANGE with PREFIX Statement

FVOLSR

The first VOLSER in the range. Specify a one- to six-character VOLSER.

LVOLSR

The last VOLSER in the range. Specify a one- to six-character VOLSER. You can specify the same value as for fvolser to have a single volume range.

R_PREF

One- to five-character prefix value that is to be substituted in the rack number passed in the L-Record to EDGCNVT.

Up to 500 VOLRANGE with PREFIX statements can be defined in SYSIN.

IF VOLRANGE WITH MEDIANAME: Use the VOLRANGE with MEDIANAME statement in following figure to assign a media name to a set of volumes based on the VOLSER prefix. The media name should be one that matches those you will define on the VLPOOL and LOCDEF DFSMSrmm parmlib options.

```
IF VOLRANGE IS fvolser TO lvolser THEN SET MEDIANAME med_name
```

Figure 69. VOLRANGE with MEDIANAME statement

FVOLSR

The first VOLSER in the range. Specify a one- to six-character VOLSER.

LVOLSR

The last VOLSER in the range. Specify a one- to six-character VOLSER. You can specify the same value as for fvolser to have a single volume range.

MED_NAME

The one- to eight-character media name that will be known to DFSMSrmm after conversion.

Up to 500 VOLRANGE with MEDIANAME statements can be defined in SYSIN if no execution parameter is specified. If more are required, you can increase the limit using the optional execution parameter.

IF VOLRANGE WITH SGNAME: Use the VOLRANGE with SGNAME statement in the following figure to assign a storage group name to a set of volumes based on the VOLSER prefix.

```
IF VOLRANGE IS fvolser TO lvolser THEN SET SGNAME sg_name
```

Figure 70. VOLRANGE with SGNAME statement

FVOLSR

The first VOLSER in the range. Specify a one- to six-character VOLSER.

LVOLSR

The last VOLSER in the range. Specify a one- to six-character VOLSER. You can specify the same value as for fvolser to have a single volume range.

SG_NAME

The one- to eight-character storage group name that will be known to DFSMSrmm after conversion.

Up to 500 VOLRANGE with SGNAME statements can be defined in SYSIN if no execution parameter is specified. If more are required, you can increase the limit using the optional execution parameter.

Note: For complete conversion of exported logical volumes you may have to run EDGUTIL with MEND(VOLUME) so that the correct DFSMSrmm information can be built into the converted records. When you run EDGUTIL with VERIFY note the messages issued and whether the stacked volume conversion is required.

IF VOLRANGE WITH TYPE: Use the VOLRANGE with TYPE statement in the following figure to assign a volume type of either LOGICAL or STACKED to a range of volumes.

```
IF VOLRANGE IS fvolsr TO lvolsr THEN SET TYPE type
```

Figure 71. VOLRANGE with TYPE statement

FVOLSR

The first VOLSER in the range. Specify a one- to six-character VOLSER.

LVOLSR

The last VOLSER in the range. Specify a one- to six-character VOLSER. You can specify the same value as for fvolsr to have a single volume range.

TYPE May be either LOGICAL or STACKED

Up to 500 VOLRANGE with TYPE statements can be defined in SYSIN if no execution parameter is specified. If more are required, you can increase the limit using the optional execution parameter.

IF VOLRANGE WITH NORACK: Use the VOLRANGE with NORACK statement in the following figure to prevent having racks assigned to a specified volume range.

```
IF VOLRANGE IS fvolsr TO lvolsr THEN SET NORACK
```

Figure 72. VOLRANGE with NORACK statement

FVOLSR

The first VOLSER in the range. Specify a one- to six-character VOLSER.

LVOLSR

The last VOLSER in the range. Specify a one- to six-character VOLSER. You can specify the same value as for fvolsr to have a single volume range.

Up to 500 VOLRANGE with NORACK statements can be defined in SYSIN if no execution parameter is specified. If more are required, you can increase the limit using the optional execution parameter.

EDITING THE VOLUME MEDIA NAME: Use this statement to change the media name, which is defined for a volume when the EDGCVNT input records contain a media name that you do not want to use.

For example, you have volumes to which the extract programs have given a media name of CART3480 and TAPE6250, but you want to use CARTS and REEL as the media names instead. You would code the statements:

IF VMEDIA EQUALS CART3480 THEN MEDIANAME EQUALS CARTS

IF VMEDIA EQUALS TAPE6250 THEN MEDIANAME EQUALS REEL

The following figure shows the statements that you can code.

```
IF VMEDIA EQUALS TAPE1600 THEN MEDIANAME EQUALS media
IF VMEDIA EQUALS TAPE6250 THEN MEDIANAME EQUALS media
IF VMEDIA EQUALS CART3480 THEN MEDIANAME EQUALS media
IF VMEDIA EQUALS CART3490 THEN MEDIANAME EQUALS media
IF VMEDIA EQUALS CART349E THEN MEDIANAME EQUALS media
IF VMEDIA EQUALS CART3590 THEN MEDIANAME EQUALS media
IF VMEDIA EQUALS CART3592 THEN MEDIANAME EQUALS media
IF VMEDIA EQUALS ANYVALUE THEN MEDIANAME EQUALS media
```

Figure 73. VMEDIA Statement with MEDIANAME

MEDIA

Media name to be assigned to all volumes with the specified media name. Media names are one to eight characters, and you must ensure that the value specified matches the VLPOOL media names that you specify in the DFSMSrmm PARMLIB.

You can specify only one IF VMEDIA statement for each VMEDIA value. If you specify more than one statement, the last value is used.

ASSIGNING A DEFAULT MEDIA NAME: Use this statement to assign a media name to any volume that does not have a media name value specified.

```
IF VMEDIA EQUALS '      ' THEN MEDIANAME EQUALS media
```

Figure 74. VMEDIA with MEDIANAME Statement

MEDIA

Media name to be assigned to all volumes that have no media name. Media names are one to eight characters, and you must ensure that the value specified matches the VLPOOL media names that you specify in the DFSMSrmm PARMLIB.

You can only specify one IF VMEDIA with MEDIANAME statement to set a default value. If you specify more than one statement, the last value is used.

ASSIGNING OPTION RM: Use the OPTION RM statement to specify the default retention method. For details of retention methods available refer to the "**z/OS DFSMSrmm Implementation and Customization Guide**" (SC26-7405). You can override the default retention method at the volume set level by setting the appropriate flags in the 'L record'. See the EDGCLREC mapping: LVRMVRS, LVRMEXP.

```
OPTION RM retention_method
```

Figure 75. OPTION RM

retention_method

one of the following:

VRSEL

Specify VRSEL to set the default retention method for all volume sets to be VRSEL. This option enables DFSMSrmm inventory management to attempt to match data sets and

volumes to VRSEs, and if a match is found, to determine if the data set or volume is to be retained by VRS. If OPTION RM is omitted retention method VRSEL is the default.

EXPDT

Specify EXPDT to set the default retention method for all volume sets to be based on EXPDT. Data sets and volumes managed by this retention method are never processed by VRSEL inventory management.

The EXPDT retention method manages at the volume level: each volume is considered separately for expiration, and each file on a volume can increment the volume expiration date.

ASSIGNING OPTION EXPDT_RETAINBY: Use the OPTION EXPDT_RETAINBY statement to specify the default Retain By for volumes managed by retention method EXPDT. This is to specify how DFSMSrmm is to retain a volume or multi volume set, as individual volumes, or as a set, or to base expiration information on the first file of the volume/volume set. For more information on the Retain By volume attribute for volumes managed by retention method EXPDT refer to the "**z/OS DFSMSrmm Implementation and Customization Guide**", Chapter 3 Retention Method. You can override the default EXPDT_RETAINBY value at the volume level by setting the appropriate value in the 'L-Record'. See the EDGCLREC mapping: LVEXPDT_RETAINBY. The Retain By value of the first volume of a multi volume set will be propagated to all other volumes of the multi volume set.

```
OPTION EXPDT_RETAINBY retainby_value
```

Figure 76. OPTION EXPDT_RETAINBY

retainby_value

one of the following:

VOLUME

The volume expiration date is set to the maximum date of the volume expiration date and the data set expiration dates on it. If OPTION EXPDT_RETAINBY is omitted VOLUME is the default.

FIRSTFILE

The expiration date of the first file is used to set the expiration date of the volume or multi volume set. All volumes in a set will have the exact same expiration date.

SET

The expiration date is the highest expiration date of all volumes and all data sets in the set. All volumes in the set will have the exact same expiration date.

EXPDT_RETAINBY is not used for data sets on retention method VRSEL managed volumes.

For more details how the expiration date of a data set or a volume is set during the conversion, refer to "EDGCNVT Considerations for Expiration Date" on page 187.

ASSIGNING OPTION EXPDT_LASTREF: Use the OPTION EXPDT_LASTREF statement to specify the default LASTREF extra days for data sets on volumes managed by retention method EXPDT. For more information on LASTREF data set attribute refer to the "**z/OS DFSMSrmm Implementation and Customization Guide**", Chapter 3 Retention Method. You can override the default EXPDT_LASTREF extra days at the data set level by setting the appropriate value in the 'L-Record' or 'D-Record'. See the EDGCLREC mapping: LDS1LRED. See EDGCDREC mapping: DSNLRED. If a data set is a multi volume data set the LASTREF extra days of the first file of the data set will be propagated to all other files of the multi volume data set.

```
OPTION EXPDT_LASTREF extra_days
```

Figure 77. OPTION EXPDT_LASTREF

extra_days

Extra_days specify the number of days that the data set will be retained after the data set was last referenced. Extra_days is a decimal number between 0 and 93000. The data set LASTREF extra_days is used to evaluate the data set expiration date. The extra_days is added to the date of last reference or in case the date of last reference is zero added to the create date. The data set expiration date is set to the maximum of the date calculated using data set LASTREF extra_days and the data set expiration date. Later when DFSMSrmm is up and running any reference to the data set by a write or read operation will re-determine the expiration date. If OPTION EXPDT_LASTREF is omitted extra_days equals 0 is the default.

EXPDT_LASTREF is not used for data sets on retention method VRSEL managed volumes.

For more details how the expiration date of a data set or a volume is set during the conversion, refer to “EDGCNVT Considerations for Expiration Date” on page 187.

EDGCNVT Output Files

Following table lists the output files created by EDGCNVT.

Table 78. EDGCNVT Output Files.

DDNAME	Description
SYSPRINT	The SYSPRINT file is a mandatory file used by the EDGCNVT utility to list all messages issued during processing.
SYSOUT	The SYSIN control statements required by EDGUTIL to create the DFSMSrmm CDS control record are written to this file. They will be used by the EDGJVERR sample job.
VRSCMDS	As each input K-Record is processed, it is converted to the equivalent RMM ADDVRS command, and the command is written to this file.
LIBLIST	EDGCNVT writes the DFSMSrmm volume and data set records to this file.
OWNLIST	EDGCNVT writes the DFSMSrmm owner records to this file.
BINLIST	EDGCNVT writes the DFSMSrmm rack and bin number records to this file.
VRSLIST	EDGCNVT writes the DFSMSrmm VRS records to this file.

SYSPRINT

The SYSPRINT file contains the messages generated during EDGCNVT processing. The messages contain statistics, information, and error details. Use the messages to determine whether processing has been successful and to follow up on any nonzero return code.

SYSOUT

At the end of processing EDGCNVT writes the EDGUTIL control statement to the SYSOUT file. The control statement is used during EDGUTIL CREATE processing to initialize the DFSMSrmm CDS control record, set the correct numbers for rack records and enable extended bin and stacked volume processing. Following figure shows what the file might contain after processing.

```

CONTROL -
  LBINNO(200) LBINFREE(63) -
  RACKNO(100213) RACKFREE(10115) -
  EXTENDED BIN(YES) STACKED VOLUME(YES)

```

Figure 78. Example of SYSOUT Contents

VRSCMDS

Each time that EDGCNVT successfully processes a K-Record, in addition to creating an DFSMSrmm VRS record, it can optionally create an ADDVRS subcommand that represents the converted policy (following figure).

```

RMM ADDVRS DSN('P$SECU.SG003*.**') NOGDG +
  CYCLES +
  COUNT( 3) +
  DELAY( 0) +
  OWNER(AAAAAAAA) +
  DELETE(1999/365) +
  LOCATION(SY ) +
  STORENUM( 2) +
  PRIORITY( 0) +
  NEXT(VRSN0001)
RMM ADDVRS NAME(VRSN0001) +
  OWNER(AAAAAAAA) +
  DELETE(1999/365) +
  LOCATION(CO ) +
  STORENUM( 1)

```

Figure 79. Example of VRSCMDS Contents

LIBLIST

The EDGCNVT input records together with modifications made by either the EDGCNVT policy and rules statements or the user exit, result in output data sets containing the following data:

From L-Records EDGCNVT produces:

- Volume information for all volumes in the library and all store locations
- Data set name information for the first data set on volumes.

From D-Records EDGCNVT produces data set name records for each data set, after the first data set, that originates on a volume.

The output records are ready for loading into the DFSMSrmm CDS.

OWNLIST

From O-Records EDGCNVT produces owner records for each owner that currently owns or could potentially own one or more volumes in the library or storage locations or VRS records.

BINLIST

From E-Records EDGCNVT produces:

- Rack information for empty racks that exist in the library
- Bin information for empty bins that exist in any store.

From L-Records EDGCNVT produces:

- Rack information for scratch and in-use volumes in the library

- Storage location bin information for those volumes that reside in any storage location.

VRSLIST

From the input K-Records EDGCNVT produces a VRS record containing details of a single retention and movement policy.

EDGCNVT Return Codes

EDGCNVT sets a return code when it ends. The following table describes the EDGCNVT return codes.

Table 79. EDGCNVT Return Codes

Return code	Description
0	<ul style="list-style-type: none"> • Normal processing.
4	<ul style="list-style-type: none"> • Loan location cannot be converted to a shelf-managed storage location. • Bin number 0 is set for a shelf-managed storage location. • The number of D-Records supplied for a volume was greater than the number expected.
8	<ul style="list-style-type: none"> • A location name was used that was not defined to the EDGCNVT SYSIN file.
1000	<ul style="list-style-type: none"> • An error occurred that must be corrected before the EDGCNVT program is rerun. See the SYSPRINT file for an explanation message.

EDGCNVT User Exit

EDGCNEXIT, a user exit called from EDGCNVT, can be used at your discretion. There are three calls to the user exit in the EDGCNVT program. The first call is when the exit is loaded and allows the exit to initialize for processing. For example, it can getmain storage, open a file, and prepare for processing. The second call is made each time a record is read from the DEXTRCT file, allowing you to modify the record if necessary. Each record is passed to the exit before EDGCNVT has processed it. Upon return from the exit, processing within EDGCNVT continues, using the updated DEXTRCT record. The third call is made at end of processing to allow the exit to issue any messages or free any getmain storage.

The exit is not part of EDGCNVT but you can write it to modify the input records before EDGCNVT begins to operate on them. A sample exit is provided by DFSMSrmm as member EDGCNEXIT of SAMPLIB.

You would use this exit, for example, to add information that the current tape management system does not provide. A specific case would be the addition of a VRS name (in the LDVRSVAL field of a library record or the DSVRSVAL field of a data set record) to be associated with a tape data set.

Other uses of this exit might be to alter volume ownership by updating the LVOWNER field of L-Records or to alter and/or specify the security level to be associated with a data set by updating the DSNCLV field of a D-Record and the LVSECLEV field of an L-Record.

Input

EDGCNEXIT is passed a parameter list. The address of the parameter list is in register 1 on entry. The format of the parameter list is:

OFFSET

VALUE

- +0 Function code. A word containing the function code.
 - 0 Initial call. No record is being passed. Use this exit call to perform any initialization that might be required. The sample exit obtains a work area for use on all exit calls.
 - 4 A record call. The address of a record is passed.

- 8 Final call. No record is passed. Use this exit call to perform any cleanup necessary. The sample exit frees the work area it obtained.
- +4 The return code. This is an output field to be set by the exit.
- +8 The address of the record passed to the exit.
- +12 A word you can use in the exit for any purpose. The word is set to zero on the initial call and is never changed by EDGCNVT. You can use it to save a work area address or remember any values you want to have from one exit call to the next.

Output

You can use the EDGCNXIT exit to update the contents of the record passed to you by means of the parameter list.

Set the return code in the parameter list to inform EDGCNVT of the processing you want performed:

Return Code

Meaning

- 0 Process the record. The exit may or may not have updated the record
- 8 Skip this record. If you skip a volume record, remember to skip the data set records that follow it for data sets on the same volume.
- 16 Invalid parameter list was passed to the exit. This return code causes EDGCNVT processing to end; no more records are processed.

EDGCNVT Messages

EDGCNVT-00000 A DEXTRACT RECORD HAS BEEN ENCOUNTERED THAT CONTAINS AN UNIDENTIFIABLE RECORD TYPE FIELD. THE RECORD APPEARS BELOW AND COLUMN 1 IS THE IDENTIFIER FIELD. IT SHOULD BE L, O, K, E OR D. *DXT_BASE*

Problem determination: where

DXT_BASE
INPUT CONVERSION RECORD

Severity: Error

Explanation: EDGCNVT has detected an invalid conversion record in the DEXTRACT input file.

System action: The job terminates with a return code 1000.

User response: Rerun the extraction process.

EDGCNVT-00005 THE NUMBER OF DATA SET NAME RECORDS SUPPLIED VIA THE DEXTRACT DATA SET IS GREATER THAN ANTICIPATED. THE DATA SET AND VOLUME INFORMATION ABOUT THE "EXTRA" DRECORD ARE SHOWN BELOW. **VOLSER:** *DSNVOL* **VSEQ:** *VSEQNUM* **LDSNCNT:** *DSCOUNT* **FILE NUMBER:** *ACCOUNT* **DATA SET SEQUENCE:** *FILENUM* **DSN:** *DSNNAME* DATA SET IGNORED AND CONVERSION CONTINUING.

Problem determination: where

DSNVOL
VOLUME SERIAL NUMBER

VSEQNUM
VOLUME SEQUENCE NUMBER

DSCOUNT
VOLUME RECORDS DATA SET COUNT FIELD

ACCOUNT
PHYSICAL FILE NUMBER

FILENUM
LOGICAL FILE NUMBER

DSNNAME
DATA SET NAME

Severity: Error

Explanation: None

System action: The job terminates with a return code 8.

User response: Rerun the extraction process.

EDGCNVT-00010 HOME LOCATION NAME CONTAINED IN L-RECORD DISPLAYED BELOW DOES NOT MATCH ANY OF THE LOCATION NAMES SUPPLIED VIA SYSIN OR "SHELF". *L VHLOC DXT_BASE*

Problem determination: where

L VHLOC
HOME LOCATION

DXT_BASE
INPUT CONVERSION RECORD

Severity: Error

Explanation: EDGCVNT is missing a LOCDEF statement in the SYSIN control file to translate the conversion record location name to a location name used in DFSMSrmm.

System action: The job terminates with a return code 1000.

User response: Check the SYSIN control file.

EDGCNVT-00015 LOCATION SPECIFIED ON A VOLRANGE STATEMENT DISPLAYED BELOW DOES NOT MATCH ANY OF THE LOCATION NAMES SUPPLIED VIA SYSIN OR "SHELF". *L VHLOC VOLRANGE DXT_BASE*

Problem determination: where

L VHLOC
HOME LOCATION

VOLRANGE
VOLRANGE STATEMENT

DXT_BASE
INPUT CONVERSION RECORD

Severity: Error

Explanation: No matching SYSIN LOCDEF statement has been found to match the location name used in a VOLRANGE statement.

System action: The job terminates with a return code 1000.

User response: Check the SYSIN control statements for consistency.

EDGCNVT-00020 VMEDIA STATEMENT *MEDNAME* SPECIFIES A VALUE PREVIOUSLY USED ON VMEDIA

Problem determination: where

MEDNAME
SUPPLIED MEDIA NAME

Severity: Warning

Explanation: None

System action: The job continues and sets a minimum return code of 4.

User response: Correct the SYSIN control statement.

EDGCNVT-00025 THE NUMBER OF STORAGE LOCATION NAMES TO BE CONVERTED TO RMM LOCAL DISASTER STORAGE LOCATION HAS EXCEEDED THREE.

Severity: Error

Explanation: You are limited to 3 disaster storage area names you can translate to DFSMSrmm LOCAL location name.

System action: The job terminates with a return code 1000.

User response: Check your SYSIN control statements.

EDGCNVT-00030 THE NUMBER OF STORAGE LOCATION NAMES TO BE CONVERTED TO RMM DISTANT DISASTER STORAGE LOCATION HAS EXCEEDED THREE.

Severity: Error

Explanation: You are limited to 3 disaster storage area names you can translate to DFSMSrmm DISTANT location name.

System action: The job terminates with a return code 1000.

User response: Check your SYSIN control statements.

EDGCNVT-00035 THE NUMBER OF STORAGE LOCATION NAMES TO BE CONVERTED TO RMM REMOTE DISASTER STORAGE LOCATION HAS EXCEEDED THREE.

Severity: Error

Explanation: You are limited to 3 disaster storage area names you can translate to DFSMSrmm REMOTE location name.

System action: The job terminates with a return code 1000.

User response: Check your SYSIN control statements.

EDGCNVT-00040 CONVRT HAS BEEN PASSED A DATA SET NAME THAT SHOULD HAVE BEEN CORRECTED IN THE ANALYSIS OF ERRORS IN THE ERROR DATA SET (ERROR DD) PRODUCED BY EDGC5LDR OR EDGCDYNM. THE ERROR DATA SET NAME AND VOLUME RECORDS ARE SHOWN BELOW. DSNREC VOLREC

Problem determination: where

DSNREC

DATA SET RECORD

VOLREC

VOLUME RECORD

Severity: Error

Explanation: None

System action: The job terminates with a return code 1000.

User response: Rerun the extraction process.

EDGCNVT-00045 THE NUMBER OF LIBRARY LOCATION NAMES TO BE CONVERTED TO RMM SHELF LIBRARY LOCATION HAS EXCEEDED THE LIMIT.

Severity: Error

Explanation: You are limited to 3 disaster storage area names you can translate to SHELF location.

System action: The job terminates with a return code 1000.

User response: Check and correct your SYSIN control statements.

EDGCNVT-00050 INCORRECT IF OWNER STATEMENT SUPPLIED VIA SYSIN: CNTRL_CARD

Problem determination: where

CNTRL_CARD
SYSIN CONTROL STATEMENT

Severity: Error

Explanation: EDGCNVT has detected a syntax error.

System action: The job terminates with a return code 1000.

User response: Correct the specified SYSIN control statement and rerun the job.

EDGCNVT-00055 INCORRECT IF VMEDIA STATEMENT SUPPLIED VIA SYSIN: CNTRL_CARD

Problem determination: where

CNTRL_CARD
SYSIN CONTROL STATEMENT

Severity: Error

Explanation: EDGCNVT has detected a syntax error.

System action: The job terminates with a return code 1000.

User response: Correct the specified SYSIN control statement and rerun the job.

EDGCNVT-00060 THE LOCDEF TYPE SUPPLIED VIA SYSIN IS NOT ONE OF ATL, MTL, BINS OR NOBINS AND CANNOT THEREFORE BE PROCESSED. THE NAME SUPPLIED IS: *loc_type*

Problem determination: where

loc_type LOCATION TYPE SUPPLIED IN THE SYSIN CONTROL STATEMENT

Severity: Error

Explanation: None

System action: The job terminates with a return code 1000.

User response: Correct the specified SYSIN control statement and rerun the job.

EDGCNVT-00065 THE VRS OPTION SUPPLIED VIA SYSIN IS INVALID. THE VRS OPTION SUPPLIED IS: CNTRL_CARD

Problem determination: where

CNTRL_CARD
SYSIN CONTROL STATEMENT

Severity: Error

Explanation: The VRS release option must be one of XDI or SDI.

System action: The job terminates with a return code 1000.

User response: Correct the specified SYSIN control statement and rerun the job.

EDGCNVT-00066 THE OPTION OPTION SUPPLIED VIA SYSIN IS INVALID. THE OPTION OPTION SUPPLIED IS: CNTRL_CARD

Problem determination: where

CNTRL_CARD
SYSIN CONTROL STATEMENT

Severity: Error

Explanation: The *OPTION* option must be *RM* with one of *VRSEL* or *EXPDT*.

System action: The job terminates with a return code 1000.

User response: Correct the specified *SYSIN* control statement and rerun the job.

EDGCNVT-00070 THE NUMBER OF *TEXT* STATEMENTS HAS EXCEEDED *TABSIZE*. INCREASE THE PROGRAM'S CAPACITY BY SPECIFYING A LARGER VALUE VIA THE *PARM* KEYWORD OF THE *JCL EXEC* STATEMENT.

Problem determination: where

TEXT ONE OF *LOCDEF*, *VOLRANGE* OR *VOLPRE*

TABSIZE
CURRENT LIMIT

Severity: Error

Explanation: *EDGCNVT* has a single, optional execution parameter to increase the maximum number of *LOCDEF*, *VOLRANGE* or *VOLPRE* statements to support. The default value is 500 of each type.

System action: The job terminates with a return code 1000.

User response: Specify a value to increase the limit and rerun the job.

EDGCNVT-00075 *CNTRL_CARD* WRITE CARD IN ERROR * *VOLRANGE* OVERLAP WITH *ENTRY_START* TO *ENTRY_END* CORRECT *VOLRANGE* INPUT AND RERUN

Problem determination: where

CNTRL_CARD
SYSIN CONTROL STATEMENT

ENTRY_START
OVERLAPPED VOLUME RANGE START VOLUME

ENTRY_END
OVERLAPPED VOLUME RANGE END VOLUME

Severity: Error

Explanation: At least two *VOLRANGE* *SYSIN* control statements were overlapping.

System action: The job terminates with a return code 1000.

User response: Check and correct the *SYSIN* control statement.

EDGCNVT-00080 *CNTRL_CARD* WRITE CARD IN ERROR * *VOLRANGE* IS INCORRECT CORRECT *VOLRANGE* INPUT AND RERUN

Problem determination: where

CNTRL_CARD
SYSIN CONTROL STATEMENT

Severity: Error

Explanation: *EDGCNVT* has detected an invalid *SYSIN* control statement where the start volume serial number is greater than the end volume serial number.

System action: The job terminates with a return code 1000.

User response: Correct the *SYSIN* control statement and rerun the job.

EDGCNVT-00085 RMMSTORE VALUE DOES NOT MATCH TO ANY LOCATION NAME PROVIDED WITH LOCDEF. STATEMENT IN ERROR IS: CNTRL_CARD

Problem determination: where

CNTRL_CARD
SYSIN CONTROL STATEMENT

Severity: Error

Explanation: None

System action: The job terminates with a return code 1000.

User response: Correct the SYSIN control statement and rerun the job.

EDGCNVT-00090 NO BIN NUMBER PROVIDED IN *LRSBIN* FIELD FOR VOLUME ASSIGNED TO A SHELF MANAGED STORAGE LOCATION. VOLUME *LVOLSER* ASSIGNED FROM *LVLOCNAM* TO HOME LOCATION *L VHLOC*

Problem determination: where

LRSBIN
CONVERSION RECORD BIN

LVOLSER
CONVERSION RECORD VOLUME SERIAL NUMBER

LVLOCNAM
CONVERSION RECORD LOCATION NAME

L VHLOC
CONVERSION RECORD HOME LOCATION

Severity: Warning

Explanation: None

System action: The job continues and sets a minimum return code of 4.

User response: Check and, if you correct the conversion input, rerun EDGCNVT.

EDGCNVT-00095 INVALID *FIELDNAME* DATE *FIELDVAL* FOUND IN *RECORDID* RECORD *RECORDKEY* INVALID DATE HAS BEEN REPLACED WITH DEFAULT DATE 1999365.

Problem determination: where

FIELDNAME
CONVERSION RECORD DATE FIELD NAME

FIELDVAL
CONVERSION RECORD DATE FIELD VALUE

RECORDID
ONE OF VOLUME, DATASET, DATASET VRS, NAME VRS, VOLUME VRS OR VRS

RECORDKEY
CONVERSION RECORD

Severity: Error

Explanation: None

System action: The job continues and sets a minimum return code of 8.

User response: Check and, if you correct the conversion input, rerun EDGCNVT.

EDGCNVT-00100 INCORRECT VRS SPECIFICATION FOUND. THE VRS IS NOT CREATED: DATA SET NAME MASK: VDSNAME JOB NAME MASK: VRSJOBN

Problem determination: where

VDSNAME
VRS DATA SET NAME

VRSJOBN
VRS JOB NAME

Severity: Error

Explanation: EDGCNVT detected invalid characters either in the data set name mask or in the job name mask. In addition to alphanumeric characters you can use %, ^ or *.

System action: The job continues and sets a minimum return code of 8.

User response: Correct the conversion input and rerun the job.

EDGCNVT-00105 CURRENT LOCATION NAME CONTAINED IN L-RECORD DISPLAYED BELOW DOES NOT MATCH ANY OF THE LOCATION NAMES SUPPLIED VIA SYSIN OR "SHELF". LOCATION NAME SET TO "SHELF". LVLOCNAM DXT_BASE

Problem determination: where

LVLOCNAM
LOCATION NAME IN THE CONVERSION RECORD

DXT_BASE
INPUT CONVERSION RECORD

Severity: Warning

Explanation: For a location name specified in the conversion input file there does no corresponding LOCDEF statement exist. The location is set to SHELF.

System action: The job continues and sets a minimum return code of 8.

User response: None

EDGCNVT-00110 THE LOCATION NAME CONTAINED IN THE K-RECORD DISPLAYED BELOW DOES NOT MATCH ANY OF THE LOCATION NAMES SUPPLIED VIA SYSIN OR "HOME". (SHELF IS NOT A VALID NEXT DESTINATION NAME.) (DO NOT USE LOAN LOCATION NAMES.) LOCATION NAME SET TO "HOME".

Severity: Warning

Explanation: For a location name specified in the conversion input file there does no corresponding LOCDEF statement exist. The location is set to the home location.

System action: The job continues and sets a minimum return code of 8.

User response: None

EDGCNVT-00115 THE LOCATION NAME CONTAINED IN THE E-RECORD DISPLAYED BELOW DOES NOT MATCH ANY OF THE LOCATION NAMES SUPPLIED VIA SYSIN OR "SHELF".

Severity: Warning

Explanation: For a location name specified in the conversion input file there does no corresponding LOCDEF statement exist.

System action: The job continues and sets a minimum return code of 8.

User response: None

EDGCNVT-00120 THE RACK NUMBER LIMIT *MAXBIN* FOR LOCATION *LOCNAM* HAS BEEN REACHED. NO FURTHER EMPTY BIN RECORDS WILL BE CREATED.

Problem determination: where

MAXBIN
RACK NUMBER LIMIT

LOCNAM
LOCATION NAME

Severity: Error

Explanation: None

System action: The job continues and sets a minimum return code of 4.

User response: None

EDGCNVT-00125 LOAN LOCATION *LONLOC* CANNOT BE CONVERTED TO A STORAGE LOCATION WITH BINS FOR VOLUME *VOLSER*

Problem determination: where

LONLOC
LOAN LOCATION

VOLSER
VOLUME SERIAL NUMBER

Severity: Warning

Explanation: A loan location cannot be a bin-managed location.

System action: The job continues and sets a minimum return code of 4.

User response: Check the conversion input and, if you correct the data set name, rerun the job.

EDGCNVT-00130 EMPTY BIN RECORD *ESSTART* IGNORED FOR LOCATION *ESLOCNAM* TYPE *LOCTYPE* MEDIANAME *EVMEDIA*

Problem determination: where

ESSTART
EMPTY BIN RECORD BIN

ESLOCNAM
EMPTY BIN RECORD LOCATION

LOCTYPE
ONE OF ATL, MTL OR NOBINS OR LOAN

EVMEDIA
EMPTY BIN MEDIA NAME

Severity: Information

Explanation: You can't have bins in an sms-managed library, on loan or in a NONBINS location.

System action: The job continues.

User response: Check the conversion input and, in case, rerun the extraction process.

EDGCNVT-00135 BIN NUMBER "000000" USED IN L-RECORD DISPLAYED BELOW IS INVALID FOR LOCATION - HOME LOCATION USED *LVLOCNAM*

Problem determination: where

LVLOCNAM
CONVERSION RECORD LOCATION NAME

Severity: Warning

Explanation: None

System action: The job continues and sets a minimum return code of 4.

User response: None

EDGCNVT-00140 EDGCNVT PROCESSING STARTED

Severity: Information

Explanation: None

System action: None

User response: None

EDGCNVT-00145 THE NUMBER OF DATA SET NAME RECORDS SUPPLIED VIA THE DEXTRACT DATA SET IS NOT AS EXPECTED. IT SHOULD BE VOLUME RECORDS DATA SET COUNT FIELD (LDSNCNT) MINUS ONE. THE VOLUME INFORMATION IS SHOWN BELOW. VOLSER: *VVOLSER* VSEQ: *VSEQNUM* LDSNCNT: *DSCOUNT* ACTUAL COUNT: *ACCOUNT* HIGHEST FILE NUMBER: *FILENUM* PREVVOL: *MVPVOL* COUNT CORRECTED AND CONVERSION CONTINUING.

Problem determination: where

VVOLSER

VOLUME SERIAL NUMBER

VSEQNUM

CONVERSION RECORD LOCATION NAMEVOLUME SEQUENCE NUMBER

DSCOUNT

VOLUME RECORDS DATA SET COUNT FIELD

ACCOUNT

PHYSICAL FILE NUMBER

FILENUM

LOGICAL FILE NUMBER

MVPVOL

PREVIOUS VOLUME SERIAL NUMBER

Severity: Warning

Explanation: None

System action: The job continues and sets a minimum return code of 4.

User response: None

EDGCNVT-00150 A DATA SET NAME CONTAINING LEADING BLANKS HAS BEEN DETECTED ON A NON SCRATCH VOLUME. DATA SET NAME HAS BEEN CLEARED AND DATA SET COUNT HAS BEEN SET TO ZERO. CONVERSION CONTINUES. VOLSER: *VOLSER* DATA SET NAME: *DSNAME*

Problem determination: where

VOLSER

VOLUME SERIAL NUMBER

DSNAME

DATA SET NAME

Severity: Warning

Explanation: EDGCNVT has detected an invalid data set name for a non-scratch single-file volume.

System action: The job continues and sets a minimum return code of 4.

User response: Check the conversion input and, if you correct the data set name, rerun the job.

EDGCNVT-00155 A DATA SET NAME WITH LEADING BLANKS HAS BEEN DETECTED ON A NON SCRATCH VOLUME CONTAINING MORE THAN ONE DATA SET. NO CORRECTIONS HAVE BEEN MADE. PLEASE CHECK AND CORRECT THE DATA SET NAME. CONVERSION CONTINUES. VOLSER: VOLSER
DATA SET NAME: DSNAME

Problem determination: where

VOLSER
VOLUME SERIAL NUMBER

DSNAME
DATA SET NAME

Severity: Error

Explanation: None

System action: The job continues and sets a minimum return code of 8.

User response: Correct the data set name and rerun the job.

EDGCNVT-00160 NO CLOSING QUOTE FOR STORLOC VALUE OR STORLOC VALUE IS MORE THAN 8 CHARACTERS. STATEMENT IN ERROR IS: CNTRL_CARD

Problem determination: where

CNTRL_CARD
SYSIN CONTROL STATEMENT

Severity: Error

Explanation: None

System action: The job terminates with a return code 1000.

User response: Correct the SYSIN control statement and rerun the job.

EDGCNVT-00165 ** WARNING ** EDGCNVT IS NOT APF AUTHORIZED NO MASK OR TDSI VALIDATION PERFORMED.

Severity: Warning

Explanation: None

System action: The job continues and sets a minimum return code of 16.

User response: EDGCNVT should reside in an authorized library and be link-edited to AC(1).

EDGCNVT-00170 THE MEDIA TYPE OR MEDIA INFORMATION NAME SPECIFIED ON IF VOLPRE IS NOT SUPPORTED. VALUE IN ERROR: ccccccc

Problem determination: where

ccccccc MEDIA TYPE OR MEDIA INFORMATION NAME VALUE

Severity: Error

Explanation: None

System action: The job terminates with a return code 1000.

User response: IF VOLPRE statement should contain either IBM media types or vendor specific media information and type defined in the EDGRMMxx member of PARMLIB. Vendor specific media type should be preceded by media information.

EDGCNVT-00175 RECORDING FORMAT AND MEDIA TYPE FOR VOLUME *vvvvvv* ARE NOT CONSIST.

Problem determination: where

vvvvvv VOLUME SERIAL NUMBER

Severity: Warning

Explanation: None

System action: The job continues and sets a minimum return code of 8.

User response: Make sure that DFSMSrmm is started in the manual mode, and the corresponding MEDINF information is specified on the MEDINF command in the DFSMSrmm PARMLIB member. For conversion records created by EDGC5LDR see the SAMPLIB member EDGMEDST for MEDINF commands that are ready to be used.

EDGCNVT-00180 Expiration date *YYYY/ddd* of volume *nnnnnn* is lowered to *YYYY/ddd*, which is the expiration date of first data set in volume set: *dsname* on volume *nnnnnn*.

Problem determination: where

YYYY/ddd

Expiration date (Julian format)

nnnnnn Volume serial number

dsname Data set name

Severity: Warning

Explanation: EDGCNVT has detected that the volume EXPDT_RETAINBY is FIRSTFILE and updates all volume expiration dates to the expiration date of the first data set in the volume set. You will see this message if the expiration date of the volume is lowered by this processing.

System action: The job continues and sets a minimum return code of 4.

User response: It is recommended to check the conversion input and, if you decide to adjust the expiration date of the volume or the expiration date of the first data set in the volume set, or you decide to change the volume attribute RETAINBY of the first volume in the set, rerun the job.

EDGCEBIN Input Files

Following table lists the input files used by EDGCEBIN:

Table 80. EDGCEBIN Input Files

DDNAME	Description
SORTBIN	This temporary file is created after running EDGCNVT (see Figure 49 on page 188 step SORT) and contains the bin records from the BINLIST file sorted by User-Store Name, User-Store Bin Media Name, and User-Store Bin Name. See "BINLIST" on page 203 for information about the BINLIST file records.

EDGCEBIN Output Files

Following table lists the output files created by EDGCEBIN.

Table 81. EDGCEBIN Output Files

DDNAME	Description
SYSPRINT	The SYSPRINT file is a mandatory file used by the EDGCEBIN program to list all messages issued during processing.
BINLIST	EDGCEBIN appends the DFSMSrmm empty bin number records to this EDGCNVT output file to fill the gaps between used bins with empty bins.

EDGCEBIN Return Codes

EDGCEBIN sets a return code when it ends. Following table describes the EDGCEBIN return codes.

Table 82. EDGCEBIN Return Codes

Return Code	Description
0	Normal processing.
4	The SORTBIN file is empty.
8	Some records of the SORTBIN file contain non-digital bin numbers, or incorrect execution parameter value specified. See the SYSPRINT file for more information.
12	Syntax error in the execution parameter. See the SYSPRINT file for more information.

EDGCEBIN Messages

EDGCEBIN-00000 INCORRECT INPUT PARAMETER: *input_param*

Problem determination: where

input_param

EDGCEBIN EXECUTION PARAMETER

Severity: Error

Explanation: Either the provided execution parameter is not syntactical correct or the prefix length number is not in the defined range.

System action: The step is ended with a return code of 12.

User response: Correct the execution parameter and restart the job.

EDGCEBIN-00005 DD SORTBIN INPUT FILE IS EMPTY

Severity: Information

Explanation: The supplied SORTBIN input file does not contain any bin records.

System action: The step is ended with a return code of 4.

User response: Check the DD card. If the BINLIST data set created *input_param* with EDGCNVT does not contain any rack or bin records there is no need to run EDGCEBIN.

EDGCEBIN-00010 NON-DIGITAL BIN NUMBER IN THE SORTBIN FILE ENTRY: *bin_record* **OR INCORRECT INPUT PARAMETER SPECIFIED:** *input_param*

Problem determination: where

bin_record

SORTBIN INPUT RECORD

input_param

EDGCEBIN EXECUTION PARAMETER

Severity: Error

Explanation: Some records of the SORTBIN file contain non-digital bin numbers, or incorrect execution parameter value specified.

System action: EDGCEBIN sets a return code of 8 and continues processing.

User response: Correct the input and rerun the program.

Creating the DFSMSrmm CDS

After the successful execution of the EDGCNVT program, you must execute some other steps in order to get an DFSMSrmm CDS ready to be used for parallel running and validation of the conversion process. The steps to accomplish this are described below.

Run EDGJLOAD

The sample job in SAMPLIB member EDGJLOAD sorts the output records from the EDGCNVT run, sorts the owners generated and removes duplicates, processes the records with the DFSMSrmm EDGCVOVL utility, finally sorting them using the DFSORT program.

The job allocates the DFSMSrmm CDS and loads the records with an IDCAMS REPRO operation.

Following figure shows the sample JCL.

```

//*JOBNAME JOB .....
/*
/* $01=      ,110,940422,MWW: make the JCL work
/*
//SORT1 EXEC PGM=SORT,REGION=4M
//NEWOWN DD DISP=(MOD,DELETE),SPACE=(TRK,1),UNIT=SYSDA,
//        DSN=STSGWD.EDGCVOVL.INTERM.DATA
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=STSGWD.EDGCNVT.LIBLIST.DATA
//SORTOUT DD DISP=(,PASS),AVGREC=K,SPACE=(1000,(70,10),RLSE),
//         DSN=&&SORTLIB,UNIT=SYSDA
//SYSIN DD *
OPTION NOCHALT
SORT FIELDS=(5,1,CH,D,213,8,CH,A,7,6,CH,A),DYNALLOC
/*
//SORT2 EXEC PGM=SORT,REGION=6M,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=STSGWD.EDGCNVT.OWNLIST.DATA
//SORTOUT DD DSN=&&SORTOWN,
//          DCB=*.SORTIN,
//          UNIT=SYSDA,DISP=(,PASS),
//          SPACE=(400,(1,1),RLSE),AVGREC=K
//SYSOUT DD SYSOUT=*
//SYSIN DD *
OPTION NOCHALT
SORT FIELDS=(6,8,CH,A),DYNALLOC
SUM FIELDS=NONE          removes duplicates
END
//CVOVL EXEC PGM=EDGCVOVL,REGION=4M,COND=(0,NE)
/**STEPLIB DD DISP=SHR,DSN=SYS1.LINKLIB
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CTLOUT DD DISP=(OLD,PASS),DSN=&&SORTLIB
//OWNOUT DD DISP=(OLD,PASS),DSN=&&SORTOWN
//NEWOWN DD DISP=(,PASS),DSN=STSGWD.EDGCVOVL.INTERM.DATA,
//        AVGREC=K,SPACE=(1000,(100,30),RLSE),UNIT=SYSALLDA
//MERGE EXEC PGM=SORT,REGION=4M,COND=(0,NE)
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=STSGWD.EDGCNVT.LIBLIST.DATA,LRECL=2044
//        DD DISP=SHR,DSN=STSGWD.EDGCNVT.BINLIST.DATA
//        DD DISP=SHR,DSN=STSGWD.EDGCVOVL.INTERM.DATA
//        DD DISP=SHR,DSN=STSGWD.EDGCNVT.VRSLIST.DATA
//SORTOUT DD DISP=(,PASS),AVGREC=K,SPACE=(1000,(100,30),RLSE),
//         DSN=&&MERGE,UNIT=SYSDA
//SYSIN DD *
OPTION NOCHALT
SORT FIELDS=(5,56,CH,A),DYNALLOC
/*

```

Figure 80. (Part 1 of 2). Sample EDGJLOAD Execution JCL

```

//DELDEF EXEC PGM=IDCAMS,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE STSGWD.RMM.CDS
SET MAXCC = 0
DEFINE CLUSTER (NAME(STSGWD.RMM.CDS) -
FILE(MASTER) -
FREESPACE(15 0) -
KEY(56 0) -
REUSE -
RECORDSIZE(512 9216) -
SHAREOPTIONS(3 3) -
KILOBYTES(4500 1500) -
VOLUMES(DFRMMA) -
DATA (NAME(STSGWD.RMM.CDS.DATA) -
BUFFERSPACE(829440) -
CISZ(26624) -
INDEX (NAME(STSGWD.RMM.CDS.INDEX) -
CISZ(2048))
/*
//LOAD EXEC PGM=IDCAMS,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//IN DD DISP=(OLD,PASS),DSN=&&MERGE
//OUT DD DISP=SHR,DSN=STSGWD.RMM.CDS,
// AMP=('BUFND=50,BUFNI=50')
//SYSIN DD *
REPRO INFILE(IN) OUTFILE(OUT)
/*

```

Figure 81. (Part 2 of 2). Sample EDGJLOAD Execution JCL

Before running this EDGJLOAD job, you can tailor some parameters, such as the space allocation values for the DFSMSrmm CDS. To choose the space allocation values you can get information from the reports you generated during the environment analysis phase. You need to know how many volumes and data sets you have, how many locations you must define, and the number of VRs. The output of the extract program can also be useful in getting the information.

After you have the information, you must calculate the amount of space required by using the space calculation rules in the "**z/OS DFSMSrmm Implementation and Customization Guide**" (SC26-7405) .

While you are loading the DFSMSrmm CDS using the IDCAMS utility, you may see:

```
IDC3314I - RECORDS OUT OF SEQUENCE
```

This message may be issued when duplicate records are loaded into the DFSMSrmm CDS. The loading of duplicate records may occur if errors were detected during the extract process. Refer back to the output produced during the extract to see whether any return codes higher than 4 were produced and ensure that all corrective actions are taken before rerunning the programs.

Note also that IDCAMS REPRO continues on for only four errors and then quits. So you may have more errors to correct than those that IDCAMS identifies.

Create DFSMSRmm CDS Control Record

The DFSMSrmm CDS control record contains information about the contents of the DFSMSrmm CDS, such as the number of shelf locations in the library and built-in storage locations.

After you have run EDGJLOAD you must create the DFSMSrmm CDS control record by using the output from the SYSOUT DD-statement of the EDGCNVT program. The SYSOUT DD-statement contains the SYSIN control statements that the EDGUTIL utility requires to create the DFSMSrmm CDS control record.

A sample job for creating the DFSMSrmm CDS control record can be found in the SYS1.SAMPLIB EDGJVERR. It is shown in following figure.

You run the job to create the DFSMSrmm CDS control record and verify the DFSMSrmm CDS that was built. Run the job as described in the next section.

Note: See "**z/OS DFSMSrmm Implementation and Customization Guide**" (SC26-7405) for more information about EDGUTIL.

Verifying the DFSMSrmm CDS

You must check the VSAM data set to ensure that all information stored in it is in the correct format, and without inconsistencies. Use the EDGUTIL utility with a job like that shown in the following figure.

Any customer that has System Managed Tape and using LOCDEF statements for EDGCNVT SYSIN DD should plan to run EDGUTIL with VERIFY(VOLCAT) to ensure that the RMM CDS built by conversion is in synch with the TCDB. They could combine the EDGUTIL VERIFY(ALL) with the VOLCAT, by using EDGUTIL with VERIFY(VOLCAT,ALL).

```
//*JOBNAME JOB .....
//CREATE EXEC PGM=EDGUTIL,PARM='CREATE'
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//MASTER DD DISP=OLD,DSN=STSGWD.RMM.CDS
//SYSIN DD DISP=SHR,DSN=STSGWD.EDGCNVT.SYSOUT.DATA
//* *****
//* SYSIN DD *
//* CONTROL MASTER(Y) RACKNO(41275) RACKFREE(0) -
//* CDSID(MYCDSID) -
//* RBINNO(3539) RBINFREE(435)
//* /*
//* *****
//VERIFY EXEC PGM=EDGUTIL,PARM='VERIFY(ALL)',
//*ERIFY EXEC PGM=EDGUTIL,PARM='VERIFY(VOLCAT,ALL)',
// COND=(0,NE),REGION=64M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//MASTER DD DISP=OLD,DSN=STSGWD.RMM.CDS,
// AMP=('BUFND=50,BUFNI=50')
//SYSIN DD DUMMY
```

Figure 82. Sample EDGUTIL Execution JCL for VERIFY Processing

You can find sample JCL in the SYS1.SAMPLIB EDGJVERR.

Refer to the "**z/OS DFSMSrmm Implementation and Customization Guide**" (SC26-7405) for more information on the parameters that you can use with the EDGUTIL program.

Review the output of the verification job carefully to see whether there are error messages that will affect the integrity of the DFSMSrmm CDS. If there are errors that do not allow you to proceed to the parallel running phase, you can try to correct the errors and go back to the beginning of the conversion phase, after the extract process, change the necessary EDGCNVT control statements, and restart the conversion steps.

If you cannot correct the errors, it may be necessary to contact an IBM service representative who can help you in the use of the EDGUTIL MEND utility, which may correct the errors.

Note: It should not be necessary to use the MEND utility regularly. Limit its use to fixing DFSMSrmm CDS problems introduced by errors in the conversion process that initially created the DFSMSrmm CDS. Please ask for IBM assistance before you use the MEND utility.

Running DFSMSrmm Commands

Once you have successfully verified the integrity of the DFSMSrmm CDS, it may be necessary to run some DFSMSrmm commands to:

- Verify and eventually add bins to storage locations
- Add VRSs if the conversion process has not added them.

Note: These commands cannot be run before starting DFSMSrmm.

You must consider preparing the command stream that adds VRSs during this phase of the migration in order to be ready for the parallel running stage.

Adding Empty Bins

Note: If you use shelf-managed storage locations, you might have been able to add empty bins during the extract process for every storage location defined, such as with EDGC5BIN ADDBIN or MAXBIN option. If you were not able to add bins, or if you are migrating from a manual tape management and do not have empty bins defined, you can add them now by using the DFSMSrmm ADDBIN command.

First you must know how many bins you need for every storage location, and then you can execute the command:

```
RMM ADDBIN bin_number LOCATION(myloc) COUNT(n_of_bins)
```

where

bin_number

is the bin number that you want to add

myloc is the location name where you add bins

n_of_bins

is the number of bins that you want to add.

Adding VRSs

The VRS conversion can be done either manually or through the conversion process. It may be easier to create the VRS definitions manually than to use the conversion programs when:

- There is a small set of VRS definitions to convert.
- The VRS was not built properly by the conversion program and does not meet your needs.
- You are using the VRSCMDS file from the EDGCNVT program.

If your installation has only a few VRS rules that can be easily defined, you can use the DFSMSrmm ADDVRS command rather than the VRS conversion process.

In addition, if the VRS rules in your previous tape management environment cannot be easily converted, such as "cycle by days" retention or some particular GDG retention methods, you must consider using the DFSMSrmm ADDVRS command to add them.

The EDGCNVT program shipped with DFSMSrmm gives you an additional opportunity to build VRS definitions by using the output generated in the VRSCMDS DD file. This method can be useful if you

want to check VRS definitions before adding them. You can review the definitions in this file and change the VRS commands as necessary before submitting the command stream.

If you decided to use the EDGUX100 user installation exit without the UXTABLE than you have to run the EDGCMD that contains RMM ADDVRS in addition to merging the ACS statements prepared in EDGACS, defining management classes and activating the ACS routines. EDGCMD and EDGACS are generated by EDGRACS in the EDGJSRDS sample JCL. (see Figure 19 on page 54)

Note: Remember to delete any previous VRS definition from the DFSMSrmm CDS before adding the VRSs from the VRSCMDS file to avoid duplication or error messages.

For more information about using the DFSMSrmm ADDVRS command, refer to the "**z/OS DFSMSrmm Managing and Using Removable Media**" (SC26-7404).

Chapter 10. Running in parallel with RMM

Exit Modifications

In order to get both tape management systems to process when running RMM in parallel with the existing system, the OAM tape exits CBRUXENT and CBRUXEJC as well as the message display exit IGXMSGEX have been enhanced to provide parallel running capability.

Exit Overview

When you run RMM in parallel with another tape management system, some exits must be used by both systems, and perhaps some are not required. When exits are used by both systems, the tape management system that is in control must be taking decisions and the second system must be tracking what is happening.

For CBRUXENT and CBRUXEJC, the parallel running changes enable your existing exits to get control before RMM and the decision taken by your existing exits is used in preference to the RMM decision. By running the RMM processing after your existing exits RMM can record any information modified by your current exits.

For IGXMSGEX, which updates tape drive displays, you only need either one system or the other to be updating the drive display. As long as you use the same scratch pool names either RMM or your existing system can be updating the drive display. During the early days of running in parallel you may want your existing system updating displays and nearer production time you may want RMM to update the displays. The parallel running change to IGXMSGEX enables your current exit to get control, but not RMM.

Activation of Parallel Running Exits

To setup this function use one of the following methods:

1. Using an SMP/E USERMOD

- Create and install an SMP/E USERMOD as shown in the figure below to provide parallel running capability.

Note: Copy the RMM-supplied IGXMSGEX to a source library and edit it if necessary. Copy the source of the IGXMSGEX exit from the existing tape management system after the end of the RMM-supplied source and change the name of the CSECT to ANOMSGEX (or other selected name). Now you have both 'exits' in a single source member each with a different name. Now install the SMP/E USERMOD and when IGXMSGEX is applied it will actually install both RMM-supplied and the parallel run exit from the existing tape management system.

If you need to edit the source code to set your own selected alternate exit load module names, copy the latest RMM source code from either SMPSTS or AEDGSR1, and alter the TXLIB to point to your source library that contains your modified source code.

```

//RMMSTUFF JOB , 'SLIP IT IN',MSGCLASS=H,MSGLEVEL=(1,1)
//STEP1 EXEC SMPPEMVS,REGION=6120K
//SMPCTL DD *
SET BDY(GLOBAL) .
RECEIVE .
/*
//SMPPTFIN DD DATA,DLM=##
++USERMOD (VMRMM03) REWORK(2000287) .
++VER (Z038) FMID(HDZ11D0) PRE(UWxxxxx) /*
Change FMID as needed.
Edit PRE as needed or use the BYPASS(PRE)
operand on the APPLY command */ .
++SRC(CBRUXENT) TXLIB(AEDGSR1) DISTLIB(AEDGSR1) /*
AEDGSR1 points to latest source level
assumption is that PTFs are already accepted
if not, use SMPSTS as the TXLIB */ .
++SRC(CBRUXEJC) TXLIB(AEDGSR1) DISTLIB(AEDGSR1) .
++SRC(IGXMSGEX) TXLIB(AEDGSR1) DISTLIB(AEDGSR1) .
##
//STEP2 EXEC SMPPEMVS,REGION=6120K
//SMPCTL DD *
SET BDY(GLOBAL) .
UCLIN .
ADD UTILITY(ASMSYSP) NAME(ASMA90)
PARM(XREF(FULL),NOOBJECT,DECK,RENT,SYSPARM(YES)) .
ADD OPTIONS(ASMSYSP)
ASM(ASMSYSP) .
ENDUCL .
/*
//STEP3 EXEC SMPPEMVS,REGION=6120K
//SMPCTL DD *
SET BDY(TGT1) OPTIONS(ASMSYSP) .
APPLY SELECT(VMRMM03) .
/*

```

Figure 83. Sample USERMOD

- To implement changes to CBRUXENT and CBRUXEJC you do not need to IPL, but can copy the updated load modules into LINKLIB and refresh LLA.
- To implement changes to IGXMSGEX you must copy IGX00030 into LPALIB and re-IPL with CLPA. You may not want to IPL at the time you switch RMM to production so careful planning of the change to IGX00030 is required. We recommend that you replace the parallel running exit for IGXMSGEX while you continue running in parallel once you are sure that RMM is providing the expected scratch pooling support.
You should have been running in warning mode and not have any error messages about volumes from incorrect scratch pools.
- When the parallel running is not needed remove the USERMOD and install the standard load modules again.

2. Outside SMP/E

- To setup the parallel running capability for the OAM tape exits CBRUXENT and CBRUXEJC you can:
 - Rename the existing exit load module names as follows:
 - CBRUXENT to ANOUXENT
 - CBRUXEJC to ANOUXEJC
 - Assemble the RMM-supplied exits with PARM='SYSPARM(YES)'.
 - Link the RMM-supplied exits.
- To setup the parallel running capability for the message display exit IGXMSGEX you can:
 - Assemble the RMM-supplied exit with PARM='SYSPARM(YES)'.

- Link the RMM-supplied exit together with the renamed IGXMSGEX exit provided by the existing tape management system. Use JCL shown in the figure below as an example:

```
//LINK1 EXEC PGM=IEWL,REGION=2M,
// PARM='LET,LIST,XREF,NCAL,RENT,NCAL'
//LPALIB DD DISP=SHR,DSN=SYS1.LPALIB
//*
//SYSLMOD DD DISP=SHR,DSN=DFRMM1.NEW.LOAD
//AEDGMOD1 DD DISP=SHR,DSN=SYS1.AEDGMOD1
//*
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLIN DD *
REPLACE EDGMSGEX
REPLACE IGXMSG01
REPLACE IGX00030
RENAME IGXMSGEX,ANOMSGEX
CHANGE IGXMSGEX(ANOMSGEX)
INCLUDE LPALIB(IGX00030)
NAME ANOMSGEX(R) RC=4
INCLUDE AEDGMOD1(IGXMSGEX)
INCLUDE LPALIB(IGX00030)
INCLUDE SYSLMOD(ANOMSGEX)
ORDER IGX00030,IGXMSG01
ORDER IGXMSGEX,EDGMSGEX
MODE RMODE(ANY)
ENTRY IGX00030
NAME IGX00030(R) RC=4
/*
```

Figure 84. Sample JCL to Link the IGXMSGEX Exit

The first step of the link-edit is to extract the existing IGXMSGEX provided by the existing tape management system from IGX00030 and rename it to ANOMSGEX, ready to be re-included for parallel running. The second step of the link-edit is to update the existing load module with the RMM supplied IGXMSGEX and add in the ANOMSGEX section.

- Alternatively you can edit the RMM-supplied exits (the figure below shows the CBRUXENT exit as an example) to set the &ANOEXIT variable with the new name of the existing exit and the &PARALLEL variable to YES and after that assemble and link the updated RMM-supplied exits.

```
CBRUXENT TITLE 'DFSMSRMM CBRUXENT SAMPLE USER EXIT'
&ANOEXIT SETC 'ANOEXIT'      Replace ANOEXIT with required name
&PARALLEL SETC '&SYSPARM'    Replace &SYSPARM with YES if req'd
      AIF ('&PARALLEL' EQ '').SETNO
      AIF ('&PARALLEL' EQ 'YES').SETYES
.SETNO ANOP
&PARALLEL SETC 'NO'
.SETYES ANOP
```

Figure 85. Code to Modify for Parallel Exit Run

Chapter 11. EDGCVRSX (sample of EDGUX100)

Overview

Introduction

This sample program includes the same functionality as the EDGUX100 sample exit. It also implements a external, dynamic table that can be used for applying retention policies based on data set and jobname, and special keyword dates, and assignment of scratch pool.

Note: We recommend that you maintain your retention policies in your SMS ACS routines instead of in the UXTABLE. To do this, run the EDGRACS exec of the EDGJSRDS sample JCL (see Figure 19 on page 54). The EDGRACS exec translates the VRS management values from the source UXTABLE generating ACS statements and placing them into the data set designated by DD name EDGACS, and also generating RMM ADDVRS commands and placing them in the data set designated by DD name EDGCMD. EDGACS contains ACS FILTLIST and WHILE-SET statements. You can update the statements based on your required policies and merge them with the existing ACS routines. EDGCMD contains RMM ADDVRS commands, for RETPD entries from source UXTABLE, that updates the RMM CDS. Before running the commands you can manually update the ADDVRS statements according to your required policies. In this case you can use the EDGUX100 samplib member as your installation exit without the UXTABLE. (see “Creating SMS ACS Statements using EDGRACS” on page 57)

Functions

The EDGCVRSX sample implements the following EDGUX100 installation exit functions:

1. 'sticky label' processing. At CLOSE/EOV you get volume details to allow support for sticky label printing. Comments are placed in the code to show where you can add this function if required.
2. Control recording of multiple files for selected applications. Using data set name, job name, and program name combinations you can specify that only the first file on a volume is recorded and managed by RMM. The sample code implements this function based on a table in the sample program. To enable the support you just have to add entries to the EDMTAB table using the sample, commented entries as a model.
3. Ability to request RMM to ignore a tape (a 'foreign' tape, or duplicate volume serial) by coding EXPDT=98000, or ACCODE=xCANORES in the job JCL.
4. Ability to assign Vital Record Specification Management values or retention period based on expiry date or data set name / job name combination specified in a dynamic table.

Note: In order to fully implement this support, a Vital Record Specification (VRS) must be defined to RMM by command for each VRS Management Value specified in the dynamic table. If no such VRS definition exists, the VRS management Value is ignored by RMM.

5. Ability to assign scratch pool based on data set name / job name combination specified in a dynamic table.
6. Special key word date support. Any special dates coded in the dynamic table are recognized and the retention policy applied. Special key word dates are those like 99000, 99ccc, and 98ddd. Any date specified in the dynamic table is treated as a special date, unless you bypass special date processing by using a special dummy ddname. The ddname is hard-coded in the sample code as 'NOTKEYD8' (not-key-date), and is defined at label KYDDNAME. This may be changed to a suitable value as required by your installation. If the sample is used as supplied, any special date processing is bypassed if there is a:

```
//NOTKEYD8 DD DUMMY
```

DD card coded in the same JCL job step as the currently processed tape data set.

7. Ability to assign a retention method via UXTABLE based on either data set name/job name combination or special keydate in the dynamic table. Ability to override the DFSMSrmm VRSEL processing for specific data sets as they are created or re-written.

The data set VRSELEXCLUDE attribute is set for all data sets on volumes managed by the EXPDT retention method, and is not affected by this support. If a data set is already retained as a vital record, the vital record attribute is reset and the retention date set to the current date.

The external, dynamic table is loaded into common storage for processing, and is anchored off a system subsystem name table entry. The subsystem name used must be different to that used for the DFSMSrmm started procedure, and is hard-coded in the EDGCSVRX code as 'UXTB'. You can customize this name, but must use the same subsystem name as a parameter for EDGCVRSX when loading the table. The table can be updated and re-loaded at any time. The program EDGCVRSX is provided to load the table, and you can use the sample procedure EDGCVRSX to run the loader program.

Prerequisites to use EDGCVRSX

In order to use the full capabilities of EDGCVRSX you need to do the following steps:

1. Modify the sample EDGCVRSX code according to your needs:
 - Update the NOTKEYD8 DD name to the DD name used in your installation.
 - Update the subsystem name to be used to anchor the dynamic table. Change it from 'UXTB' to another 4 character value only if the value UXTB is already in use. Do not use the same subsystem name as used for DFSMSrmm started procedure. If you do change the subsystem name, also update the default subsystem name parameter for the EDGCSVRP procedure.

Note: The value of variable SSNAME in EDGCVRSX must be identical to the value of parameter SSN in sample procedure EDGCVRSX.

- Add entries to the EDMTAB table for any applications which should only have the first file recorded by RMM.
 - Add 'sticky label' support if required.
2. Rename EDGCVRSX to EDGUX100.
 3. Assemble and link the new EDGUX100 and place it into a library defined by LNKLISTxx. When you assemble DFSMSrmm installation exits the IBM-supplied default assembler options should be used, unless noted otherwise. Ensure that the ALIGN option is in effect.

Alternatively you can install the new EDGUX100 exit using SMP/E as described in the "**z/OS DFSMSrmm Implementation and Customization Guide**" (SC26-7405).

4. Create the dynamic table which consists of entries used to set VRS management value, retention period or scratch pool, based on selection by data set name and jobname, or special date. Optionally assign a retention method, or set the VRSELEXCLUDE attribute for data sets created at open or re-written. You can do this by modifying the sample table created during conversion job EDGJSRDS. EDGJSRDS creates a table called UXTABLE. The remainder of this description refers to UXTABLE but you can use EDGCVRSX instead. EDGCVRSX is a single part table while UXTABLE is in 3 parts - the 2nd of which is built by the conversion programs.
5. Compile and link UXTABLE. (See sample EDGJSRDS for the supplied, sample JCL or use sample below). The library you link it to should be identified by the STEPLIB DD in the EDGCVRSX procedure.

```

/**
/** FOLLOWING STEPS ASSEMBLES AND LINKAGE EDITS THE UXTABLE.
/** THE UXTABLE CONSISTS OF 3 PARTS:
/** 1. EDGCVRSF PREDEFINED START OF UXTABLE
/** 2. EDGCVRSX **** MACRO DEFINITIONS ****
/** 3. EDGCVRSY PREDEFINED END OF UXTABLE
/**
/** EXEC HLASM
/**C.SYSIN DD DISP=SHR,DSN=&RMLLIB(EDGCVRSF)
/** DD DISP=SHR,DSN=EDGCVRSX MACRO DEFINITION ENTRIES <-----

```

```

//          DD DISP=SHR,DSN=&RMMLIB(EDGCVRSE)
//C.SYSLIB DD DISP=SHR,DSN=SYS1.MACLIB
//          DD DISP=SHR,DSN=SYS1.MODGEN
//          DD DISP=SHR,DSN=&RMMLIB
//LINKEDIT EXEC PGM=IEWL,PARM='LIST'
//SYSLMOD DD DSN=SYS1.RMM.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSLIN  DD DSN=&&OBJ,DISP=(OLD,DELETE)
//          DD *
          NAME UXTABLE(R)
//*
```

6. Load it into common storage by using sample procedure EDGCVRSP.

7. Refresh LLA

```
F LLA,REFRESH
```

8. Refresh RMM exit module EDGUX100

If a previous version of the exit is already active, issue:

```
SETPROG EXIT,REPLACE,EXITNAME=EDG_EXIT100,MODNAME=EDGUX100
```

Alternatively, or when the exit is not active, use

```
SETPROG EXIT,DELETE,EXITNAME=EDG_EXIT100,MODNAME=EDGUX100
```

```
SETPROG EXIT,ADD,EXITNAME=EDG_EXIT100,MODNAME=EDGUX100
```

See "**z/OS MVS System Commands**" (SA22-7627) for details on the SETPROG command

How to reload the dynamic table

If you have new or changed retention requirements and therefore need to change the dynamic table do the following:

1. Modify the table according to your needs.
2. Compile and link it. (Refresh LLA may be needed)
3. Reload the table by using sample procedure EDGCVRSP.

Note: If you choose to use a different subsystem name then the old table will not be freed from common storage area.

The EDGCVRSX sample finds the dynamic table by using the hard-coded subsystem name, UXTB by default, and validates that the loaded table is the expected format. If you change the subsystem name at load time, remember that the hard-coded name in the exit must also be changed and the exit re-loaded.

The dynamic table is validated by checking the eyecatcher at the head of the dynamic table.

Rules how EDGCVRSX finds a match in dynamic table

1. During processing EDGCVRSX scans the dynamic table to find up to 2 matching table entries. One is a match on data set name and job name combination. One is a match to a special JCL EXPDT or ACCODE value. ACCODE values are converted to the special date equivalent in order to match to the KEYDATE entries in the table.

Note: Flag JFCBEXP is used to distinguish dates set via EXPDT= in the JCL from those set by RETPD=. Only dates set via EXPDT are tested to see if there is a match in the dynamic Table.

Any EXPDT support is bypassed if a **//NOTKEYD8 DD DUMMY** (or the DD name you changed it to) DD card is coded in the job step JCL.

2. Do not specify both data set name and keydate on a single table entry. Because in this case the matching routine checks only for keydate match, so, a match on data set name / job name would never be found.

3. When both dsn/job name and keydate entries match the RO (retention override) value of the dynamic table entry matching to dsn/job name is used to decide which of the two matching dynamic table entries is used to set the VRS value etc.. If the RO value is 'YES' the values of the entry matching to dsn/job name entry is used. If the RO value is 'NO' the values of the entry matching to the keydate is used.

I know this isn't easy to understand therefore the same again for those who understand pseudo code better.

```
IF data set name / job name match AND keydate match THEN
  IF dsn/job name entry includes RO=YES THEN
    Use values of dynamic table entry matching to dsn/job name
    to set VRS value, extra retention period etc.
  ELSE
    Use values of dynamic table entry matching to keydate
    to set VRS value, extra retention period etc.
  END
END
```

4. The sequence of DSN/jobname table entries is very important. The matching routine does not scan the whole table for best match but stops searching on first match found. Example:

```
Table entry 1: DSN=STRAUS*      VRSVAL=CATALOG POOL=STR*  RO=YES
                RM=EXPDT
Table entry 2: DSN=STRAUS.TEST* VRSVAL=STRAUS POOL=TEST* RO=YES
                RM=EXPDT
Table entry 3: DSN=HENICKE.VX.* VRSVAL=CATALOG POOL=MZ*   RO=YES
                RM=VRSEL VX=YES
Table entry n: .....
```

DSN in JCL = STRAUS.TEST.VRSVALUE

In this example the values from table entry 1 will be used for pooling, VRS value, and Retention Method although table entry 2 is a more specific match.

Table entry 3 shows an example to exclude a subset of data sets from VRSEL processing specifying VX=YES. For VX=YES the VRS management value set by VRSVAL=CATALOG is ignored. When DFSMSrmm honours your request to exclude a data set from VRSEL, processing ensures that the data set vital record attribute is reset and the retention date is set to current date. The matching VRS info is left unchanged.

Examples:

Assume current date is 01/01/1999

Table 83. Dynamic table sample (UXTABLE)

DSN	Jobname	Keydate	RO	Pool	VRS value	Extra Retention Period
TEST.*	TESTJOB		YES	A0001*	LDATE001	1
WOOD.*	WOODJOB		NO	A0002*	LDATE002	2
GOHR.TEST*			YES	A0003*	LDATE003	3
GOHR.*			NO	A0004*	LDATE004	4
*			NO	A0*	CATALOG	3
		99010	YES		CYCLE010	10
		99020	YES		CYCLE020	20
		99030	YES		CYCLE030	30
		99040	YES		CYCLE040	40

Table 84. Sample JCL parameter & values assigned from EDGCVRSX based on above table.

Test Case	DSN	Jobname	RETPD	EXPDT	Matching row number(s) from above sample table	Assigned Pool	Assigned VRS value	Volume Expiration Date
1	TEST.ONE	TESTJOB	11		1	A0001*	LDATE001	01/02/1999
2	TEST.TWO	TESTJOB		99010	1,6	A0001*	LDATE001	01/02/1999
3	TEST.THREE	TEST3	12		5	A0*		01/13/1999
4	TEST.FOUR	TEST4		99010	5,6	A0*	CYCLE010	01/11/1999
5	GOHR.TEST5	GOHRJOB		99030	3,8	A0003*	LDATE003	01/04/1999
6	STRAUS.SIX	JOB8			5	A0*	CATALOG	01/04/1999

Test case 1:

There is only one match in the table. It contains retention override (RO) = YES. This means JCL values will be overridden by table values.

Test case 2:

There are two matches. Because the DSN/jobname match contains RO = YES, the values from the DSN/jobname match overrides JCL values.

Test case 3:

There is only one match in the table. It contains RO = NO. Therefore only the pool value will be used.

Test case 4:

There are two matches. The DSN/jobname match contains RO=NO and the keydate match contains RO=YES. So the pool value of the DSN/jobname is assigned and the JCL values are overridden by the keydate entry.

Test case 5:

There are two matches. Both contain RO=YES. So, the DSN/jobname match takes precedence and those values override JCL values.

Test case 6:

There is only one match. It contains RO=NO. But because there is neither RETPD nor EXPDT specified in JCL the values of the matching entry will be used.

Messages

EDGI100I SUBSYSTEM UXTB NOT DEFINED OR NO TABLE LOADED

Severity: Information

Explanation: The error occurred in FIND_TAB routine while the system was trying to find the dynamic table.
Possible causes:

1. No dynamic table is loaded
2. Subsystem name used in EDGCVRSX differs from subsystem name used while loading the dynamic table (Parm SSN in EDGCVRSX)

System action: Processing continues with limited functions.

User response: Ensure that the subsystem names used in EDGUX100 and the EDGCVRSX procedure are the same, and re-load the table using EDGCVRSX.

EDGCVRSG (Macro)

Function

This macro can be used to build entries in the dynamic table. EDGJSRDS creates the 2nd part of UXTABLE by building EDGCVRSG macro statements from the RDS. EDGCVRST, the single part sample table can be customized with multiple EDGCVRSG created entries. The EDGCVRSG macro statements are also used by EDGRACS in the EDGJSRDS sample JCL to build SMS ACS statement and RMM ADDVRS commands.

Invocation

```
EDGCVRSG KEYDATE=' ',DSN=' ',JOB=' ',VRSVAL=' ',POOL=' ',           x
          DSECT=NO,RETPD=0,ACLOPT=0,RO=YES,RM=NONE,VX=NONE
```

Parameter description

The following keywords are available for the EDGCVRSG macro in the EDGUX100 installation exit:

ACLOPT

You can use the ACLOPT keyword to disable the cartridge loader for this request.

- 1 Cartridge loader is disabled for this request
- 0 Cartridge loader is enabled. This is the normal processing. This is the default value.

DSECT

YES/NO Specifies whether to produce a map for a table entry. NO is the default value.

DSN One to forty four characters, following MVS data set naming conventions, including % and *.

- % can be used to ignore a positional character in the data set name.
- * can be used to ignore all remaining characters in the data set name. A **DSN=*** means that the entry applies to all data sets.

The use of * is not the same as in the generic data set names supported by RMM for vital record specifications and search data set masks. Here the * works like the *.* might in a generic data set name mask.

Either KEYDATE, or DSN and JOB may be specified.

JOB One to eight alphanumeric or national characters including % and *.

- % can be used to ignore a positional character in the job name
- * can be used to ignore all remaining characters in the job name. A **JOB=*** means that the entry applies to all jobs.

Either KEYDATE, or DSN and JOB may be specified.

KEYDATE

Specifies an EXPDT value, or an ACCCODE date equivalent value used in JCL. A 5 character numeric value, such as 99000, 99003, 98012.

Either KEYDATE, or DSN and JOB may be specified.

POOL Specifies the RMM VLPOOL prefix to be assigned to all matching data sets. When a POOL is specified you can also specify ACLOPT.

RO Specifies the retention override option.

YES (Default) , means to override any specified JCL RETPD or EXPDT value. NO, means to accept the JCL value. When YES is specified, any matching data set has the expiration date and management value set using the current table entry values. If a data set matches to an entry with

RO=NO, EDGUX100 continues to scan the table for any KEYDATE values that match the JCL specified EXPDT or ACCODE special values so that it can be converted to a VRS management value.

RO value has no meaning for the KEYDATE keyword, when EXPDT= is specified in the JCL, because all keyword expiration dates are cleared and are set using the RETPD keyword value from the table. If no value is specified for RETPD, the RMM PARMLIB default retention period value will be used by RMM. When ACCODE is specified in the JCL, the RO=YES option can be used to override any JCL specified RETPD or EXPDT value.

VRSVAL

Specifies the name of a vital record specification management value. A vital record specification management value can be 1 to 8 alphanumeric or national characters. The value corresponds to the data set name used in a VRS DSNNAME policy, such as DSN('CATALOG').

RM Specifies retention method.

For each new volume set created during OCE you can select any valid retention method using installation exit EDG_EXIT100.

One of: NONE, EXPDT, VRSEL.

VX You can use the VX keyword to exclude data sets from VRSEL processing. Use VX=YES to assign the data set attribute VRSELEXCLUDE to data sets at open created or re-written.

Note: The **POOL**, **ACLOPT**, **RM** and **VX** keywords will not be used by a CA-1 conversion. You can add them later if you wish to use EDGUX100 to control scratch sub-pooling using data set names.

EDGCVRSL (UXTABLE loader)

Function

This Program can be used to load the linked UXTABLE into common storage area. If an existing UXTABLE is found to be loaded for the same subsystem name, the existing table is deleted from common storage.

Invocation

```
//LOADUXTB EXEC PGM=EDGCVRSL,REGION=4096K,PARM='X123,Y1234567'
```

X123 The subsystem name to be used to anchor the table when it is loaded. The value you specify must also be defined in EDGUX100 (EDGCVRSX) installation exit. Length must be 4 characters.

Y1234567

The load module name of the table to be loaded. UXTABLE or EDGCVRST customized table. It should contain EDGCVRSG macro statements to define options for EDGUX100 to assign to data sets. Length must be between 1 and 8 characters.

Messages

EDGCVRSL-00000 LOAD OF TABLE xxxxxxxx FAILED

Explanation: Invocation of LOAD macro to load table xxxxxxxx into CSA resulted in an error.

EDGCVRSL-00005 DATA TABLE LOADED O.K.

Explanation: Data table has been successfully loaded into CSA.

EDGCVRSL-00010 LOADER DETECTED BAD PARM

Explanation: Invalid invocation parameter length or no comma separating the two parameters.

EDGCVRSL-00020 FAILED TO ENQUEUE RMM EDGUX100.TABLE

Explanation: System exclusive enqueue on majorname=RMM minorname=EDGUX100.TABLE had a return code > 0. ENQ processing failed.

EDGCVRSL-00025 FAILED TO DEQUEUE RMM EDGUX100.TABLE

Explanation: Dequeue on majorname=RMM minorname=EDGUX100.TABLE had a return code > 0. DEQ processing failed.

EDGCVRSL-00030 FAILED TO CREATE SUBSYSTEM XXXX

Explanation: Call to IEFSSI to dynamically create a new subsystem name table entry failed. Probably the subsystem you specified is not dynamic. Do not specify the name of a subsystem defined in IEFSSNxx.

EDGCVRSL-00035 DO NOT USE THE RMM SUBSYSTEM NAME OR ANY OTHER ALREADY IN USE

Explanation: Subsystem name DFRM has been used or the subsystem is already in use by another program.

User response: Use the correct subsystem name.

EDGCVRSL-00040 FAILED TO UPDATE SUBSYSTEM XXXX

Explanation: Call to IEFSSI to update SUBDATA2 field had a return code > 0. This means that the new table has not been loaded correctly so it is not available for use by EDGUX100.

EDGCVRSL-00045 LOADER DETECTED BAD TABLE XXXXXXXX

Explanation: The loader did not find the eyecatcher UXTABLE in the specified load module.

EDGCVRSP (Procedure to start EDGCVRSL)

Function

This Procedure can be used to execute EDGCVRSL and pass the necessary parameters to it to get the dynamic table loaded.

You can use the following keywords when starting EDGCVRSP procedure:

TAB Is the load module name for the EDGUX100 table. it should contain EDGCVRSG macro statements to assign retention options for EDGUX100 to assign to data sets. Length must be between 1 and 8 characters. The default value is UXTABLE. For example:

```
S EDGCVRSP,TAB=EDGCVRST
```

SSN Is the subsystem name to be used to anchor the table when it is loaded. the value you specify must also be defined in EDGUX100 installation exit. Length must be 4 characters. The default value is UXTB. For example:

```
S EDGCVRSP,TAB=EDGCVRST,SSN=DYNT
```

The following command loads the UXTABLE for the UXTB subsystem to be used by EDGUX100. The default values are used.

```
S EDGCVRSP
```

The sample procedure contains a STEPLIB. The STEPLIB is an optional DD-statement. It identifies the load module library that contains the table to be loaded.

Appendix A. Modification Required for EPIC/MVS and RMM Parallel Run

```
//STSGMALO JOB  , 'AVIS.M RMM OCE LNK', CLASS=S, USER=STSGMA,
//  NOTIFY=STSGMA, MSGCLASS=X, COND=(4,LT)
/*JOBPARM  R=SS,L=20
/** THIS JOB PROVIDES AND INSTALLS A BYPASS TO ENABLE RMM AND EPIC
/**   TO RUN IN PARALLEL.
/** UPDATED FOR EPIC/MVS 3.2
/** UPDATED FOR UW90355,UW32712,UW32713,UW32922
/**
/** WHY THE BYPASS IS REQUIRED
/**   Without this bypass, if you have EPIC/MVS (Prevail/XP Media)
/**   installed on the system at the same time as you have RMM started
/**   with EDGSSSI in IEFSSNxx, or have started RMM in a mode other
/**   than MANUAL, tape usage will be affected.
/**   EPIC does not need to be active to result in these problems;
/**   if the TSID entry is in IEFSSNxx that counts as installed.
/**   Tape usage is affected by:
/**   - In warning and protect modes, RMM issues no messages about
/**     incorrect tapes mounted and cannot validate tape usage.
/**     This is because EPIC bypasses RMMs validation.
/**   - A loop in EOVS processing can occur
/**
/**   If you do not install the bypass, you must IPL with EPIC
/**   removed from the system in order to have fully functional RMM
/**   tape processing. With the bypass installed you can more easily
/**   move from EPIC to RMM.
/**
```

Figure 86. JCL to Modify RMM for EPIC/MVS Parallel Run (Part 1 of 11)

```

/** INSTRUCTIONS:
/** It is not advisable to install this bypass using SMP/E.
/** Although the ZAP can be installed under SMP/E, the link-edit
/** required for IFG0194A MUST be performed outside of SMP/E or the
/** incorrect link-edit will be done.
/**
/** 1) Use IEBCOPY or ISPF/PDF 3.3 to save a copy of your current
/**    LPALIB data set. Specify this data set name in the LPASAVE
/**    DD in the LK job step.
/**    If you rerun this job, you should not need to repeat the
/**    saving of the original LPALIB.
/** 2) If you have DFSMSrmm V1.1 installed you should update the
/**    ZAP5E and ZAP4A job steps to use the alternative, commented-out
/**    ZAP. The current ZAPs are correct for all other levels
/**    of DFSMSrmm and DFRMM R2.
/**    OFFSETS MAY NEED TO BE UPDATED FOR LATEST RMM MAINTENANCE.
/**    IF THE ZAPS DO NOT FIT, REVIEW THE COMMENTS IN THE AMASPZAP
/**    CONTROL STATEMENTS. IF YOU STILL NEED HELP, CALL SUPPORT CENTER
/** 3) Update the SYSLMOD DD in LK job step to
/**    identify the LPALIB you are installing the bypass on.
/** 4) Update the SYSLIB DD in AMASPZAP job steps to
/**    identify the LPALIB you are installing the bypass on.
/** 5) Update the SYSLIB DD in AMBLIST job step to
/**    identify the LPALIB you are installing the bypass on.
/** 6) Tailor the job card and submit the job
/** 7) Use the AMASPZAP DUMPT and AMBLIST output to verify that
/**    the bypass is implemented correctly.
/** 8) Ensure that the SYSLIB DD statement in the EPIC startup procedure
/**    identifies the LPALIB updated by this install job.
/** 9) Should the ZAP steps not complete with return code zero,
/**    review the comments in the control statements for AMASPZAP.

```

Figure 86. JCL to Modify RMM for EPIC/MVS Parallel Run (Part 2 of 11)

```

/**
/** VALIDATION OF BYPASS
/** If you wish to see trace messages for the front-end to verify
/** that it works correctly, see the assembler source for the
/** front-end and remove the * comment from the WTO macro calls.
/** If the validation fails you may want to activate the tracing
/** to identify the problem.
/** To activate RMM and EPIC/MVS tracing you can do the following:
/** 1.Active RMM trace, with operator command.
/**   F DFRMM,DEBUG=8000
/**   to turn off trace
/**   F DFRMM,NODEBUG
/** 2.Active EPIC/MVS messages using parmlib option
/**   OPTION MSG05=YES,MSG28=YES,MSG33=YESNO,MSG47=YES,MSG51=YES
/**   to turn off messages revert to your existing options.
/**
/** 1) IPL with CLPA using the new LPALIB created by this install job.
/** 2) Start RMM in WARNING or PROTECT mode.
/**   REPLY 'RETRY' to message EDG0103D if it is issued.
/**   Start EPIC.
/** 3) Submit a job that creates a new tape data set on a scratch
/**   tape. Select a tape which is not scratch status in RMM, which
/**   can be safely used for output. Mount the tape.
/**   RMM should issue a reject message.
/**   In WARNING mode, RMM will allow the tape to be used and
/**   will update the RMM CDS with information about the new tape
/**   data set. Use RMM LV volser command to list the volume details.
/**   In PROTECT mode, RMM does not allow the volume to be used.
/**   In any case, EPIC should have recorded the latest details for the
/**   new tape file created.
/**   This verifies that the change to IFG0194A was successful.
/** 4) Submit a job that creates a new tape data set on a SL tape,
/**   that is known to RMM.
/**   Mount an acceptable tape. No error messages should be issued.
/**   The job should end as normal.
/**   Verify that both RMM and EPIC have details of the data set and
/**   volume you used. Use the RMM LV command or the RMM ISPF dialog.
/**   This verifies that the change to IGC0005E was successful.

```

Figure 86. JCL to Modify RMM for EPIC/MVS Parallel Run (Part 3 of 11)

```

//*
//ASM      EXEC PGM=IEV90,REGION=5M,TIME=5,
//          PARM=('DECK,NOOBJECT,NORLD,BATCH,XREF(SHORT)',
//          'LINECOUNT(55)')   LINECOUNT=55 IS DISTRIBUTION DEFAULT
//SYSPRINT DD SYSOUT=(*,FE4A)
//SYSUT1  DD UNIT=SYSDA,SPACE=(CYL,(3,1))
//SYSPUNCH DD UNIT=VIO,SPACE=(TRK,(1,1)),BLKSIZE=800,DISP=(,PASS),
//          DSN=&&FE4A
//SYSLIB  DD DCB=BLKSIZE=32000,DISP=SHR,DSN=SYS1.MACLIB
//          DD DISP=SHR,DSN=SYS1.MODGEN
//SYSIN   DD *
194A      TITLE 'A FRONT END TO IFG0194A FOR RMM TO COEXIST WITH EPIC'
IFG0194A CSECT
IFG0554A EQU  IFG0194A
IFG0204A EQU  IFG0194A
          ENTRY IFG0554A
          ENTRY IFG0204A
          BASR 3,0           DUMP BASE REG.
          USING *,3         DECLARE ADRESSABILITY
          USING FORCORE,4   DECLARE ADRESSABILITY
          LA 1,WTOZ9        POINT AT WTO FOR Z9 CALLED
          L 11,=V(IFG019Z9) LOAD Z9 ADDRESS
          CLC TZ9,6(6)      IFG019Z9 CALLED?
          BE WTOTCALL       YES, DO PROCESSING
          L 11,=V(EDG0194A) LOAD 4A ADDRESS
          CLC T4#,6(6)      IFG0194# CALLED?
          BE CALL4A         YES, DO PROCESSING
          LA 1,WTOUNKN      POINT AT WTO FOR UNKOWN CALLED
          CLC =C'4A',6(6)   IFG0194A CALLED?
          BNE WTOTCALL      NO, THEN SELECT IFG0194A
          LA 1,WTO4ANC      POINT AT IFG0194A, R14 ^ 12
          C 14,CON12        R14 = 12
          BNE WTODCALL      NO, THEN SELECT IFG0194A

```

Figure 86. JCL to Modify RMM for EPIC/MVS Parallel Run (Part 4 of 11)

```

SPACE 2
L 15,DXUCBADR GET THE UCB
CLI UCBTBYT3-UCBCMSEG(15),UCB3TAPE CALLED FOR TAPE
BNE NOTTAPE NO, THEN SKIP
L 11,=V(IFG019Z9) LOAD Z9 ADDRESS
LA 1,WT04AZ9 POINT AT WTO FOR Z9 TO BE CALLED
MVC 6(2,6),TZ9 SET WTG TABLE
B WTOTCALL NO, THEN SKIP
SPACE 2
CALL4A LA 1,WT04# POINT AT WTO FOR Z9 CALLING 4A
MVC 6(2,6),T4A SET WTG TABLE
B WTOTCALL NO, THEN SKIP
SPACE 2
NOTTAPE LA 1,WT04ANT POINT AT WTO FOR 4A CALLED
B WTODCALL CONTINUE PROCESSING
SPACE 2
WTOTCALL DS 0H CONTINUATION ADDRESS FOR TAPE CALLS
* WTO MF=(E,(1)) ISSUE SELECTED WTO
*** REMOVE ASTERISKS FROM WTO STATEMENT TO ENABLE TRACING
LR 15,11 COPY SELECTED MODULE ADDR
BR 15 AND CONTINUE
SPACE 2
WTODCALL DS 0H CONTINUATION ADDRESS FOR DASD CALLS
**** WTO MF=(E,(1)) ISSUE SELECTED WTO
*** REMOVE ASTERISKS FROM WTO STATEMENT TO ENABLE TRACING
LR 15,11 COPY SELECTED MODULE ADDR
BR 15 AND CONTINUE
SPACE 3

```

Figure 86. JCL to Modify RMM for EPIC/MVS Parallel Run (Part 5 of 11)

```

LTORG ,
SPACE 2
PRINT NOGEN
CON12 DC F'12'
WTOZ9 WTO MF=L,ROUTCDE=2,DESC=4, X
      'IFG0194A FRONT-END DETECTED CALL TO IFG019Z9'
WTOUNKN WTO MF=L,ROUTCDE=2,DESC=4, X
      'IFG0194A FRONT-END DETECTED CALL TO UNKNOWN'
WT04# WTO 'IFG0194A FRONT-END DETECTED CALL TO IFG0194A BY IFG019Z9', X
      MF=L,ROUTCDE=2,DESC=4
WT04ANT WTO MF=L,ROUTCDE=2,DESC=4, X
      'IFG0194A FRONT-END DETECTED CALL TO IFG0194A, NOT TAPE'
WT04ANC WTO MF=L,ROUTCDE=2,DESC=4, X
      'IFG0194A FRONT-END DETECTED CALL TO IFG0194A, R14 ^ 12'
WT04AZ9 WTO 'IFG0194A FRONT-END DETECTED CALL TO IFG0194A, Z9 SELECTED',
      ,MF=L,ROUTCDE=2,DESC=4
SPACE 2
TZ9 DC C'Z9'
T4# DC X'F4C0'
T4A DC C'4A'
SPACE 2
IECDSECS (MAIN),EXPAND=YES
IEFUCBOB
END
/*

```

Figure 86. JCL to Modify RMM for EPIC/MVS Parallel Run (Part 6 of 11)

```

/*
//LK      EXEC PGM=LINKEDIT,REGION=2048K,
//          PARM='MAP,LIST,RENT,LET,NCAL'
//SYSPRINT DD  SYSOUT=(*,,LK01)
//SYSUT1  DD  UNIT=SYSDA,SPACE=(CYL,(2,2))
//LPASAVE DD  DISP=SHR,DSN=STSGMA.SAVECOPY.LPALIB
//FE4A    DD  DISP=(OLD,DELETE),DSN=&&FE4A
//SYSLMOD DD  DISP=SHR,VOL=SER=8E5SY1,UNIT=3380,DSN=SYS1.LPALIB #
//SYSLIN  DD  *
INCLUDE   FE4A
INCLUDE   LPASAVE(IFG0194A)
CHANGE    IFG0194A(EDG0194A)
INCLUDE   LPASAVE(IFG0194A)
ALIAS     IFG0194F
ALIAS     IFG0194J
ALIAS     IFG0194K
ALIAS     IFG0195G
ALIAS     IFG0204A
ALIAS     IFG0204J
ALIAS     IFG0234J
ALIAS     IFG0554A
ALIAS     IFG0554C
ALIAS     IFG0554F
ALIAS     IFG0554J
ALIAS     IFG0554K
ALIAS     IGG0190A
ALIAS     IGG0550P
ENTRY     IFG0194A
NAME      IFG0194A(R)
/*

```

Figure 86. JCL to Modify RMM for EPIC/MVS Parallel Run (Part 7 of 11)

```

INCLUDE   LPASAVE(IGC0005E)
EXPAND    IFG0552Z(24)
ALIAS     IFG0551F
ALIAS     IFG0551L
ALIAS     IFG0551N
ALIAS     IFG0552X
ALIAS     IFG0553F
ALIAS     IFG0554P
ALIAS     IFG0554T
ALIAS     IGC0003A
ALIAS     IGG0550B
ALIAS     IGG0550D
ALIAS     IGG0550F
ALIAS     IGG0550H
ENTRY     IGC0005E
NAME      IGC0005E(R)

```

Figure 86. JCL to Modify RMM for EPIC/MVS Parallel Run (Part 8 of 11)

```

/*
//ZAP4A EXEC PGM=AMASPZAP
//SYSPRINT DD SYSOUT=(*,ZAPM)
//SYSLIB DD DISP=SHR,DSN=SYS1.LPALIB,UNIT=3380,VOL=SER=8E5SY1
//SYSIN DD *
DUMPT IFG0194A IFG019Z9
NAME IFG0194A IFG019Z9
* The X'F4C1' to be changed is the first entry in the table
* created by the XCTLTABL MACRO.
* The MACRO starts with a LTORG which means that it will start
* with the double word aligned string
* CL8'IFG019Z9'
* followed by a VCON and other data including the string C'4F'.
* The next data is double word aligned, so that the C'4F' may be
* followed by some bytes of garbage, possible X'0000'.
* The target double word aligned string starts
* DC CL3'019',CL2'4A',VL3(IFG0194A)
* and it is the A of the C12'4A' which must be changed to X'C0'
* Following ZAP is for JDZ11B4 DFSMSrmm 1.2 (UW32712)
* and JDZ11C0 DFSMSrmm 1.3 (UW90355)
* and HDP3RM2 DFRMM R2 (UW32922)
VER 02A0 C9C6C7F0F1F9E9F9 =CL8'IFG019Z9'
VER 02AC E9F9 =CL2'Z9'
VER 02AE F4C6 =C'4F'
VER 02B8 F0F1F9 IHB0017A DC 0D'0',CL3'019'
VER 02BB F4C1 IDZ94A DC CL2'4A'
VER 02C0 F0F1F9 DC CL3'019'
VER 02C3 F4D1 IDZ94J DC CL2'4J'
REP 02BB F4C0 Change IDZ94A to X'F4C0'
* Following ZAP is for JDZ1150 DFSMSrmm 1.1 (UW32713)
* VER 0258 C9C6C7F0F1F9E9F9 =CL8'IFG019Z9'
* VER 0264 E9F9 =CL2'Z9'
* VER 0266 F4C6 =C'4F'
* VER 0268 F0F1F9 IHB0017A DC 0D'0',CL3'019'
* VER 026B F4C1 IDZ94A DC CL2'4A'
* VER 0270 F0F1F9 DC CL3'019'
* VER 0273 F4D1 IDZ94J DC CL2'4J'
* REP 026B F4C0 Change IDZ94A to X'F4C0'
DUMPT IFG0194A IFG019Z9
/*

```

Figure 86. JCL to Modify RMM for EPIC/MVS Parallel Run (Part 9 of 11)

```

//*
//ZAP5E EXEC PGM=AMASPZAP
//SYSPPRINT DD SYSOUT=(*,ZAPM)
//SYSLIB DD DISP=SHR,DSN=SYS1.LPALIB,UNIT=3380,VOL=SER=8E5SY1
//SYSIN DD *
  DUMPT IGC0005E IFG055Z2
  NAME IGC0005E IFG055Z2
* The patch area will be the 24 bytes of hex zeros at the end of
* the module. To reach it, use the fact that register 3 contains
* the address of the 2nd half-word in the module.
* Parts of the patch code vary, other than due to the relative
* position of the area itself. For example,
* the address of the VL3(IFG0551T) loaded by the ICM instruction.
* This can be located by the double-word aligned string
* CL5'0551T'
* which is followed by the VL3(IFG0551T).
* The CL5'0551T' will be before another double word aligned string
* which starts with X'FF' and is followed by the FMID of the product.
* The area containing the 0551T string is generated by the
* XCTLTABL MACRO which should be towards the end of the original
* module.
* Following ZAP is for JDZ11B4 DFSMSrmm 1.2 (UW32712)
* and JDZ11C0 DFSMSrmm 1.3 (UW90355)
* and HDP3RM2 DFRMM R2 (UW32922)
VER 0002 47F0,3138 BRANCH ROUND ID
VER 02A8 0000,0000,0000,0000 CHECK OFFSET OF PATCH AREA
VER 02B0 0000,0000,0000,0000 CREATED BY LINKAGE EDITOR
VER 02B8 0000,0000,0000,0000 EXPAND OPTION
VER 0238 C9C6C7F0F5F5E9F2 =CL8'IFG055Z2'
VER 0244 E9F2 =CL2'Z2'
VER 0250 F0F5F5F1E3 DC CL5'0551T',AL3(IFG0551T)
VER 0258 C9C6C7F0F5F5E9F2 =CL8'IFG055Z2'
REP 0002 47F0,32A6 BRANCH TO PATCH AREA
REP 02A8 D501,6006,3242 CLC 6(2,6),C'Z2'
REP 02AE 4780,3138 BE PAST EYECATCHER
REP 02B2 1FFF SLR 15,15
REP 02B4 BFF7,3253 ICM 15,7,ADDRESS OF IFG0551T
REP 02B8 07FF BR 15

```

Figure 86. JCL to Modify RMM for EPIC/MVS Parallel Run (Part 10 of 11)

```

* Following ZAP is for JDZ1150 DFSMSrmm 1.1 (UW32713)
* VER 0002 47F0,3138 BRANCH ROUND ID
* VER 0258 0000,0000,0000,0000 CHECK OFFSET OF PATCH AREA
* VER 0260 0000,0000,0000,0000 CREATED BY LINKAGE EDITOR
* VER 0268 0000,0000,0000,0000 EXPAND OPTION
* VER 01F0 C9C6C7F0F5F5E9F2 =CL8'IFG055Z2'
* VER 01FC E9F2 =CL2'Z2'
* VER 0200 F0F5F5F1E3 DC CL5'0551T',AL3(IFG0551T)
* VER 0208 C9C6C7F0F5F5E9F2 =CL8'IFG055Z2'
* REP 0002 47F0,3256 BRANCH TO PATCH AREA
* REP 0258 D501,6006,31FA CLC 6(2,6),C'Z2'
* REP 025E 4780,3138 BE PAST EYECATCHER
* REP 0262 1FFF SLR 15,15
* REP 0264 BFF7,3203 ICM 15,7,ADDRESS OF IFG0551T
* REP 0268 07FF BR 15
DUMPT IGC0005E IFG055Z2
/*
/**
//MAPMDL EXEC PGM=AMBLIST,PARM='LINECNT=48'
//SYSPRINT DD SYSOUT=(*,MAPM)
//SYSLIB DD DISP=SHR,DSN=SYS1.LPALIB,UNIT=3380,VOL=SER=8E5SY1
//SYSIN DD *
LISTLOAD OUTPUT=XREF,MEMBER=(IFG0194A)
LISTLOAD OUTPUT=XREF,MEMBER=(IGC0005E)
/*

```

Figure 86. JCL to Modify RMM for EPIC/MVS Parallel Run (Part 11 of 11)

Appendix B. User Survey

Conversion to RMM Process Customer Critique

Customer Name: _____

Date: _____

Converted from: _____

Customer representative completing critique: _____

Return to: or use FAX: 1 - (408) 256-0488

IBM Corporation

5600 Cottle Road

D34/050

San Jose, CA 95193

ATTN: RMM Coordinator

Overall Satisfaction Rating

1. Did the value of the process meet your expectations? ____ (Y/N)
2. What is your overall satisfaction rating with the process?
__ Delighted
__ Very satisfied
__ Satisfied
__ Dissatisfied
3. If your answer to the previous question is less than delighted, what would it take to improve your satisfaction?

4. Does the process that you've just used provide sufficient value that you would recommend its use by others? ____ (Y/N)
If not, what improvements would you require to make such a recommendation?

5. Are you willing to recommend that your company become a reference account for the process that you used? ____ (Y/N)

Note: All calls from other interested companies would be pre-screened by your IBM account team and only those deemed pertinent would be passed on to the person you designate as our focal point.

If your answer to the above question is "no", please identify your reasons.

Data Extraction Phase Assessment

If you have used the sample data extraction code, please complete this section.

Data Extraction

1. Did the execution of the data extraction code interfere with your operation? ____ (Y/N)
If so, how? _____
2. Was it necessary for you to make modifications to the supplied sample code in order to achieve your desired result? ____ (Y/N)
If so, for what purpose? _____

3. Were the directions in conversion Redbooks and this User's Guide on the use of the tools clear and understandable or did you need help in interpreting them? _____(clear/needed help)

Delivery

Check the column below that best describes your feelings about the data extraction documentation and code:

Table 85. Data Extraction Phase Assessment

	Delighted	Very Satisfied	Satisfied	Dis-Satisfied
Effectiveness				
Communications				
Technical Information				

Comments/Suggestions:

Extracted Data Conversion Phase Assessment

If you used the EDGCNVT conversion code, please complete this section.

Data Conversion

1. Did the execution of the data conversion code interfere with your operation? ____ (Y/N)
If so, how? _____
2. Was it necessary for you to make modifications to the supplied code in order to achieve your desired result? ____ (Y/N)
If so, for what purpose? _____

3. Were the directions in conversion Redbooks and this User's Guide on the use of the tool clear and understandable or did you need help in interpreting them? _____(clear/needed help)

Delivery

Check the column below that best describes your feelings about the conversion phase documentation and code:

Table 86. Extracted Data Conversion Phase Assessment

	Delighted	Very Satisfied	Satisfied	Dis-Satisfied
Effectiveness				
Communications				
Technical Information				

Comments/Suggestions:

Post-processing (CDS Creation) Phase Assessment

If you have used the conversion sample JCL, please complete this section.

CDS Creation

1. Did the execution of the post-processing code interfere with your operation? ____ (Y/N)
 If so, how? _____
2. Were the directions in conversion Redbooks and this User's Guide on the use of the JCL clear and understandable? _____(clear/needed help)

Delivery

Check the column below that best describes your feelings about the conversion phase documentation and code:

Table 87. Post-processing (CDS Creation) Phase Assessment

	Delighted	Very Satisfied	Satisfied	Dis-Satisfied
Effectiveness				
Communications				
Technical Information				

Comments/Suggestions:

Quality and Value

Please Check the column below that best describes the quality of each of the phases and their deliverables that you selected.

Data Extraction

Table 88. Quality and Value Extraction Assessment

	Excellent	Good	Average	Poor
Extraction Tools				
User's Guide				
Conversion Redbooks				
Technical Value				
Presentation of Messages				
Explanation of Messages				

Comments/Suggestions:

Data Conversion

Table 89. Quality and Value Conversion Assessment

	Excellent	Good	Average	Poor
Conversion Tools				
User's Guide				
Conversion Redbooks				
Technical Value				
Presentation of Messages				
Explanation of Messages				

Comments/Suggestions:
