IAM

INNOVATION ACCESS METHOD
IAM USER MANUAL VERSION 9.2

INNOVATION DATA PROCESSING is pleased to present the IAM User Manual.

PURPOSE OF THE GUIDE

This Users Guide outlines the capabilities, describes the basic features of these capabilities, and provides directions on their use. It was written primarily for technical personnel who have responsibilities related to the design and implementation of systems that will use IAM. This manual is intended to give those personnel a reference that will help them use IAM as a simple, efficient, reliable alternative to VSAM KSDS, ESDS, and optionally AIX and RRDS processing in batch and online applications.

INNOVATION DATA PROCESSING

IAM®, IAM/AIX, and IAM/PLEX are proprietary program products of:
INNOVATION DATA PROCESSING
Innovation Plaza
275 Paterson Avenue
Little Falls, New Jersey 07424-1658

It is available for license exclusively from INNOVATION DATA PROCESSING

Reproduction of this User Manual is prohibited except by licensed users for their internal use.

A copy of this manual is available in Adobe PDF format on the INNOVATION DATA PROCESSING Documentation CD-ROM and on the INNOVATION DATA PROCESSING FTP site for online use and local printing.

© Copyright 1996, 2013, 2015
© INNOVATION DATA PROCESSING

USA Contacts

<table>
<thead>
<tr>
<th>Corporate Headquarters</th>
<th>Voice</th>
<th>973-890-7300</th>
<th>Sales</th>
<th><a href="mailto:sales@fdrinnovation.com">sales@fdrinnovation.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fax:</td>
<td>973-890-7147</td>
<td>Support:</td>
<td><a href="mailto:support@fdrinnovation.com">support@fdrinnovation.com</a></td>
</tr>
<tr>
<td></td>
<td>Home page:</td>
<td><a href="http://www.innovationdp.fdr.com">http://www.innovationdp.fdr.com</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### European Offices

<table>
<thead>
<tr>
<th>Country</th>
<th>INNOVATION DATA PROCESSING</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>France</strong>*</td>
<td>191 Avenue Aristide Briand</td>
<td>(33) 1 49 69 94 02</td>
<td>(33) 1 49 69 90 98</td>
</tr>
<tr>
<td></td>
<td>94230 Cachan</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sales: <a href="mailto:frsales@fdrinnovation.com">frsales@fdrinnovation.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Support: <a href="mailto:frsupport@fdrinnovation.com">frsupport@fdrinnovation.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Netherlands (Nordic Countries)</strong>*</td>
<td>Brouwerstraat 8</td>
<td>036-534 1660</td>
<td>036-533 7308</td>
</tr>
<tr>
<td></td>
<td>1315 BP Almere</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sales: <a href="mailto:nlsales@fdrinnovation.com">nlsales@fdrinnovation.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Support: <a href="mailto:nlsupport@fdrinnovation.com">nlsupport@fdrinnovation.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Germany</strong>*</td>
<td>Orleansstraße 4a</td>
<td>089-489 0210</td>
<td>089-489 1355</td>
</tr>
<tr>
<td></td>
<td>D-81669 München</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sales: <a href="mailto:desales@fdrinnovation.com">desales@fdrinnovation.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Support: <a href="mailto:desupport@fdrinnovation.com">desupport@fdrinnovation.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>United Kingdom</strong>*</td>
<td>Clarendon House</td>
<td>0208-905 1266</td>
<td>0208-905 1428</td>
</tr>
<tr>
<td></td>
<td>125 Shenley Road</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Borehamwood, Herts, WD6 1AG</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sales: <a href="mailto:uksales@fdrinnovation.com">uksales@fdrinnovation.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Support: <a href="mailto:uksupport@fdrinnovation.com">uksupport@fdrinnovation.com</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

## 1 IAM VERSION 9.2 INTRODUCTION
- 01.00 IAM Version 9.2 Introduction ............................................. 1-1
- 01.01 Summary of Modifications and Enhancements ............................ 1-4
- 01.02 Prior Version 9 Summary of Modifications and Enhancements ......... 1-7
- 01.03 Organization of This Manual ............................................. 1-12
- 01.04 Control Statement Format ................................................ 1-14
- 01.05 Documentation Notation Format ........................................ 1-16
- 01.20 Vendor Products That Have Converted VSAM Files to IAM ............. 1-17
- 01.21 Vendor Products That Work With IAM .................................. 1-18

## 2 IAM CONCEPTS AND FACILITIES
- 02.00 IAM Overview .......................................................... 2-1
- 02.10 IAM Alternate Index Support ......................................... 2-4
- 02.20 IAM/PLEX: IAM Sysplex Record Level Sharing ....................... 2-6
- 02.30 IAM Record Level Sharing ............................................. 2-10
- 02.40 IAM's Real Time Tuning ............................................... 2-14
- 02.50 IAM Data Compression ................................................ 2-20
- 02.60 IAM Journaling ......................................................... 2-22
- 02.70 SMS Support in IAM .................................................. 2-23
- 02.80 Automatic DASD Space Functions .................................... 2-26
- 02.90 Other Features of IAM ................................................ 2-29

## 3 IAM DATA SET STRUCTURE
- 03.00 IAM Data Set Structure Overview ................................... 3-1
- 03.10 Enhanced Format File Structure ..................................... 3-3
- 03.11 Prime Related Overflow ................................................ 3-7

## 10 IAM VERSION 9.2 USERS GUIDE
- 10.00 IAM Users Guide Introduction ....................................... 10-1
- 10.10 JCL Considerations ..................................................... 10-3
- 10.20 Defining IAM Files .................................................... 10-5
- 10.25 Using Hardware Compression ........................................ 10-46
- 10.30 Loading IAM Datasets .................................................. 10-51
- 10.40 Processing IAM Datasets ............................................... 10-60
- 10.45 CICS Considerations ................................................... 10-74
- 10.50 IAM Tuning Guidelines ................................................ 10-85
- 10.51 IAM Storage Usage .................................................... 10-98
- 10.60 Using IAM Alternate Index Support ................................ 10-103
- 10.61 Defining an IAM Alternate Index ................................... 10-106
- 10.62 Building an IAM Alternate Index .................................. 10-117
- 10.63 Defining an IAM Path ................................................. 10-121
- 10.64 Accessing Data Through an IAM Alternate Index .................. 10-124
- 10.65 Special Considerations with IAM Alternate Indexes ............... 10-125
- 10.70 IAM Reports ............................................................. 10-127
- 10.71 IAMPRINT Report ...................................................... 10-128
- 10.72 IAMINFO Report ....................................................... 10-138
- 10.73 IAMSMFVS Reports .................................................... 10-147
- 10.74 Reporting on IAM Datasets with FDREPORT ......................... 10-152
- 10.75 IAMXMON: IAM Execution Monitor for CICS ....................... 10-155
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.80</td>
<td>IAM Dataset Maintenance</td>
<td>10-157</td>
</tr>
<tr>
<td>10.81</td>
<td>Reorganizing IAM Datasets</td>
<td>10-158</td>
</tr>
<tr>
<td>10.82</td>
<td>Deleting IAM Datasets</td>
<td>10-169</td>
</tr>
<tr>
<td>10.83</td>
<td>Renaming IAM Datasets</td>
<td>10-171</td>
</tr>
<tr>
<td>10.84</td>
<td>Recataloging IAM Datasets</td>
<td>10-176</td>
</tr>
<tr>
<td>10.85</td>
<td>Backing up and Restoring IAM Datasets</td>
<td>10-181</td>
</tr>
<tr>
<td>10.86</td>
<td>Moving IAM Datasets</td>
<td>10-192</td>
</tr>
<tr>
<td>10.87</td>
<td>Recovering IAM Datasets</td>
<td>10-195</td>
</tr>
<tr>
<td>10.88</td>
<td>IAM Journal and Recovery</td>
<td>10-205</td>
</tr>
<tr>
<td>10.89</td>
<td>Adding Volumes to an IAM Dataset</td>
<td>10-214</td>
</tr>
</tbody>
</table>

## 20 USING IAM RECORD LEVEL SHARING

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.00</td>
<td>IAM Record Level Sharing</td>
<td>20-1</td>
</tr>
<tr>
<td>20.02</td>
<td>Multiple IAM/RLS Address Spaces</td>
<td>20-7</td>
</tr>
<tr>
<td>20.03</td>
<td>IAM/RLS Implementation</td>
<td>20-10</td>
</tr>
<tr>
<td>20.05</td>
<td>Setting up IAM/RLS (Record Level Sharing)</td>
<td>20-12</td>
</tr>
<tr>
<td>20.06</td>
<td>IAM/RLS Record Lock Recovery</td>
<td>20-16</td>
</tr>
<tr>
<td>20.10</td>
<td>IAM/RLS Parameters</td>
<td>20-20</td>
</tr>
<tr>
<td>20.20</td>
<td>IAM/RLS Automatic Dataset Eligibility</td>
<td>20-28</td>
</tr>
<tr>
<td>20.30</td>
<td>IAM/RLS Journaling</td>
<td>20-34</td>
</tr>
<tr>
<td>20.40</td>
<td>IAM/RLS Operator Commands</td>
<td>20-40</td>
</tr>
<tr>
<td>20.41</td>
<td>IAM/RLS Tracing</td>
<td>20-44</td>
</tr>
<tr>
<td>20.50</td>
<td>IAM/RLS CICS Considerations</td>
<td>20-47</td>
</tr>
<tr>
<td>20.60</td>
<td>IAM/RLS Batch Considerations</td>
<td>20-54</td>
</tr>
<tr>
<td>20.70</td>
<td>IAM/RLS Data Recovery</td>
<td>20-57</td>
</tr>
</tbody>
</table>

## 25 USING IAM/PLEX RECORD LEVEL SHARING

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.00</td>
<td>IAM/PLEX Record Level Sharing</td>
<td>25-1</td>
</tr>
<tr>
<td>25.03</td>
<td>IAM/PLEX Implementation</td>
<td>25-10</td>
</tr>
<tr>
<td>25.05</td>
<td>IAM/PLEX JCL Procedure</td>
<td>25-12</td>
</tr>
<tr>
<td>25.06</td>
<td>IAM/PLEX Record Lock Recovery</td>
<td>25-16</td>
</tr>
<tr>
<td>25.10</td>
<td>IAM/PLEX Parameters</td>
<td>25-19</td>
</tr>
<tr>
<td>25.20</td>
<td>IAM/PLEX Automatic Dataset Eligibility</td>
<td>25-27</td>
</tr>
<tr>
<td>25.30</td>
<td>IAM/PLEX Journaling</td>
<td>25-34</td>
</tr>
<tr>
<td>25.40</td>
<td>IAM/PLEX Operator Commands</td>
<td>25-38</td>
</tr>
<tr>
<td>25.41</td>
<td>IAM/PLEX Tracing</td>
<td>25-42</td>
</tr>
<tr>
<td>25.50</td>
<td>IAM/PLEX CICS Considerations</td>
<td>25-45</td>
</tr>
<tr>
<td>25.60</td>
<td>IAM/PLEX Batch Considerations</td>
<td>25-52</td>
</tr>
<tr>
<td>25.70</td>
<td>IAM/PLEX Data Recovery</td>
<td>25-55</td>
</tr>
</tbody>
</table>

## 30 IAM 9.2 OVERRIDES

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.01</td>
<td>IAM Override Statements Overview</td>
<td>30-1</td>
</tr>
<tr>
<td>30.02</td>
<td>IAM CREATE Override Statement Format</td>
<td>30-3</td>
</tr>
<tr>
<td>30.03</td>
<td>IAM ACCESS Override Statement Format</td>
<td>30-14</td>
</tr>
<tr>
<td>30.04</td>
<td>IAM Override Statement Examples</td>
<td>30-24</td>
</tr>
</tbody>
</table>

## 40 IAMSMFVS - DATA SET ANALYSIS PROGRAM

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.01</td>
<td>IAMSMFVS - Data Set Analysis Program Overview</td>
<td>40-1</td>
</tr>
<tr>
<td>40.02</td>
<td>IAMSMFVS JCL Requirements</td>
<td>40-2</td>
</tr>
<tr>
<td>40.03</td>
<td>IAMSMFVS - REPORT Command</td>
<td>40-3</td>
</tr>
<tr>
<td>40.04</td>
<td>IAMSMFVS Usage Examples</td>
<td>40-7</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS

40.05 IAMSMFVS Summary Report .............................. 40-9
40.06 IAMSMFVS EXCP Reports ............................... 40-10
40.07 IAMSMFVS Data Set Summary Report ................. 40-12
40.08 IAMSMFVS Size Reports ............................... 40-14

41 SMF BASIC REPORTING UTILITY
41.01 IAMSMF Overview ........................................ 41-1
41.02 IAMSMF JCL Requirements ............................ 41-2
41.03 IAMSMF - COPY Command ............................. 41-3
41.04 IAMSMF - IAMINFO Command ......................... 41-7
41.05 IAMSMF - PRINT Command ........................... 41-15
41.06 IAMSMF - QUERY Command ........................... 41-18
41.07 IAMSMF - REPORT Command ......................... 41-23

45 IAMRECVR – IAM DATASET RECOVERY
45.01 IAMRECVR – Recovery Program Overview ............ 45-1
45.02 IAMRECVR - JCL Requirements ........................ 45-3
45.03 IAMRECVR - APPLY Command ........................ 45-4
45.04 IAMRECVR – DECOMPRESS Command ................. 45-7
45.05 IAMRECVR - DIAGNOSE Command .................... 45-9
45.06 IAMRECVR – IAMSTRUCTURE Command .......... 45-10
45.07 LIST Command .......................................... 45-11
45.08 IAMRECVR - PRINT Command ........................ 45-12
45.09 IAMRECVR - RECOVER Command .................... 45-14

46 IAMISPF – IAM’S ISPF INTERFACE
46.01 IAMISPF Overview ...................................... 46-1

47 IAMJREST - IAM JOURNAL RESTORE PROGRAM
47.01 IAMJREST – IAM Journal Restore Program Overview . 47-1
47.02 IAMJREST – JCL Requirements ....................... 47-4
47.03 IAMJREST – RESTORE Command .................... 47-5
47.04 IAMJREST Examples .................................. 47-9
47.05 IAMJREST – RELEASE Command .................... 47-11

48 IAMJUTIL – IAM JOURNAL UTILITY PROGRAM
48.01 IAMJUTIL Program Overview .......................... 48-1
48.02 IAMJUTIL – JCL Requirements ...................... 48-2
48.03 IAMJUTIL – COPY Command ......................... 48-3
48.04 IAMJUTIL – DUMP Command ......................... 48-8
48.05 IAMJUTIL – MVSLOG Command ..................... 48-13
48.06 IAMJUTIL – SCAN Command ......................... 48-15

80 IAM VERSION 9.2 MESSAGES AND CODES
80.01 IAM Version 9.2 Messages and Codes ............... 80-1
80.02 IAM WTO Messages .................................. 80-2
80.03 IAM SYSPRINT Messages ............................ 80-32
80.04 IAM Journal Exit WTO Messages ................. 80-54
80.05 IAM/RLS and IAM/PLEX Messages ................. 80-57
80.10 IAM ABEND Codes .................................... 80-133
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>80.20</td>
<td>IAM Catalog Return Codes</td>
<td>80-135</td>
</tr>
<tr>
<td>80.21</td>
<td>IAM I/O Request Error Codes</td>
<td>80-138</td>
</tr>
<tr>
<td>80.22</td>
<td>IAM Open and Close Error Codes</td>
<td>80-141</td>
</tr>
<tr>
<td>80.23</td>
<td>COBOL File Status Codes with IAM</td>
<td>80-143</td>
</tr>
<tr>
<td>90</td>
<td>IAM INSTALLATION INSTRUCTIONS</td>
<td></td>
</tr>
<tr>
<td>90.01</td>
<td>IAM Installation Introduction</td>
<td>90-1</td>
</tr>
<tr>
<td>90.02</td>
<td>Electronic Installation</td>
<td>90-2</td>
</tr>
<tr>
<td>90.03</td>
<td>CD Installation</td>
<td>90-19</td>
</tr>
<tr>
<td>90.10</td>
<td>Setting the IAM Global Options</td>
<td>90-20</td>
</tr>
<tr>
<td>90.11</td>
<td>Installing the IAM ISPF Dialog</td>
<td>90-25</td>
</tr>
<tr>
<td>90.12</td>
<td>Installing the IAM CICS Monitor</td>
<td>90-27</td>
</tr>
<tr>
<td>90.20</td>
<td>Activating the IAM VSAM Interface</td>
<td>90-30</td>
</tr>
<tr>
<td>90.30</td>
<td>Testing IAM</td>
<td>90-41</td>
</tr>
<tr>
<td>90.31</td>
<td>IAM Installation Questions</td>
<td>90-44</td>
</tr>
<tr>
<td>90.40</td>
<td>Moving a New Version of IAM into Production</td>
<td>90-46</td>
</tr>
<tr>
<td>91</td>
<td>IAM GLOBAL OPTION CHANGE FACILITY</td>
<td></td>
</tr>
<tr>
<td>91.01</td>
<td>IAMZAPOP – Overview and JCL</td>
<td>91-1</td>
</tr>
<tr>
<td>91.02</td>
<td>IAMZAPOP – Format and Options</td>
<td>91-3</td>
</tr>
<tr>
<td>91.03</td>
<td>IAMZAPOP – ZAP Command</td>
<td>91-4</td>
</tr>
<tr>
<td>91.04</td>
<td>IAMZAPOP JCL Examples</td>
<td>91-15</td>
</tr>
</tbody>
</table>
IAM Version 9.2 Introduction

What IAM IS

IAM is a reliable high performance information storage and retrieval software product that can be used in place of VSAM data sets for batch, TSO and online processing. IAM offers a level of performance and reduction in the use of computer system resources that provides substantial savings for most applications that utilize VSAM data sets. IAM reduces CPU utilization, reduces the physical I/O (EXCP’s), and offers improved utilization of DASD space that yields reductions in elapsed time for application processing.

The IAM product is different from many other VSAM performance enhancement products. Rather than trying to tweak VSAM to work faster the IAM approach starts off by using a different underlying file structure and data organization. This innovation coupled with performance oriented algorithms and coding techniques results in significant proven performance benefits.

The IAM product is being used to achieve the following benefits:

- Reduction of computing costs
- Reduction of CPU time
- Reduction of EXCP’s
- Reduction of DASD space utilization
- Reduce nightly batch processing window
- Delay processor and DASD upgrades

The IAM product is available as a base product that supports KSDS and ESDS type of files, with two optional feature packages available at extra cost that provide additional functionality. The IAM/AIX feature provides alternate index support and RRDS support. The IAM/PLEX feature provides Sysplex record level sharing support. A free trial version of IAM is available for users to evaluate the IAM product which includes the IAM/AIX optional feature. If requested it will also include the IAM/PLEX optional feature.

IAM

The base IAM product provides a unique set of features and functions to enhance and improve the performance on VSAM KSDS and VSAM ESDS types of files. These key features and functions include:

- A transparent VSAM interface that requires no changes to application programs, and generally no changes to JCL or CICS regions.

- IAM files can easily exceed 4 gigabytes. A single IAM file has a maximum size of 54 terabytes. There are no special size considerations for KSDS files. For ESDS files to exceed 4 gigabytes, they must be defined with either an 8-byte RBA (IAM XESDS) or with the PSEUDORBA feature of IAM due to the VSAM addressing architecture. The current actual size limitation is due to the z/OS DD (TIOT) restriction of a maximum of 59 volumes. For example, on 59 3390-54 volumes, the maximum is 3,186 (54g*59) gigabytes.

- IAM data sets are typically allocated as non-VSAM type of datasets, generally DSORG=PS. IAM datasets can also be defined as Large Sequential datasets (DSNTYPE=LARGE) or as a DFSMS Extended Format datasets to utilize more than 64K tracks per volume. IAM files can reside on EAV volumes with IAM.
Version 9.0 and z/OS 1.11 or higher as DFSMS Extended Format datasets, and with z/OS 1.12 as Large Format Sequential or Basic Sequential datasets. IAM files that are DSORG=DA should be deleted and redefined to use DSORG=PS.

- IAM's automatic Real Time Tuning offers dynamic buffer management techniques that are unsurpassed. With the Real Time Tuning TURBO mode, to respond faster to high I/O needs, and increased defaults for the number of buffers the need for manual tuning is eliminated for almost all files.

- Data Compression facility that can be used for all types of IAM data sets, with less CPU overhead than is typical of VSAM data compression techniques. IAM supports both an efficient proprietary software data compression technique or hardware compression. Hardware compression can be used with either IAM generating a compression dictionary during file load, or with a customized hardware compression dictionary. Users may be able to achieve even greater data compression than can be achieved with standard VSAM hardware compression when they opt to build a customized dictionary as described in this manual. Data compression also helps to reduce physical I/O required to process the data sets, because more data is contained in a physical block than when it is uncompressed.

- Record level sharing within a single z/OS image (LPAR), for concurrently executing batch jobs, CICS regions and TSO users.

- Support for journaling and recovery that offers capabilities normally seen only with data base software.

- Automatic space release after a file is initially loaded and dynamic secondary space adjustments.

- Software data compressed files can be reorganized with FDRREORG or IDCAMS without decompressing and recompressing the data. This reduces CPU time and I/O to perform the reorganizations, and it can reduce media usage when the sequential copy of the data is placed on DASD.

- A “Data in Virtual” feature called Dynamic Tabling, which keeps those records that are frequently required in a Data Space or in a virtual storage table within the users address space.

- IAM’s run time report IAMINFO, which fully describes each IAM data set processed by a job step, along with a complete processing profile, including resource usage. This data can also be captured by SMF, with reporting available from programs provided with IAM, or customers can write their own SMF analysis and reporting programs to include the IAM provided data.

IAM provides this unique set of features that are not found in other individual products. There are several software packages offered by other vendors that will provide some of the above functions, but only IAM puts all of this capability together in one package.

IAM/AIX

The IAM/AIX feature is an additional cost feature to the base IAM software product. IAM/AIX uses the full functionality and features described above on IAM base clusters with IAM alternate indexes and IAM paths and on RRDS types of files. Please note that all datasets involved in the alternate index sphere must be either all IAM datasets or all VSAM datasets. For example, you cannot have an IAM base cluster with a VSAM alternate index or a VSAM base cluster with an IAM alternate index. It typically only takes a few base clusters with alternate indexes for this feature to more than pay for itself.
IAM Version 9.2 Introduction

IAM/PLEX

The IAM/PLEX additional cost feature provides record level sharing support of IAM datasets across multiple LPARs within a z/OS Sysplex. IAM/PLEX builds upon IAM/RLS by utilizing the XCF Sysplex services and the Sysplex Logger to provide a record level sharing capability for IAM files within a Sysplex. IAM’s implementation is an evolutionary step forward from the basic single system IAM/RLS that utilizes the IAM/RLS framework that has been in use for several years. The IAM/PLEX design is different from VSAM/RLS and is intended to enable users to take advantage of record level sharing without the complexity involved with setting up a VSAM/RLS environment. IAM/PLEX includes support for sharing between online (CICS/TS) regions and batch jobs.

IAM Version 9.2 Requirements

IAM Version 9.2 utilizes various functions and features only available in z/OS operating systems and instructions available only on processors that support z/Architecture. The minimum requirements for IAM Version 9.2 are z/OS operating system and a processor that supports the fifth edition of z/Architecture Principle of Operations (-04 level) or above. The specific processors are the z/890 and z990 and above.

IAM support on systems that do not meet those requirements and are using z/Architecture machine can use up through Version 9.1. Support is provided for systems without z/Architecture by IAM Version 8.1.

Due to quad-word alignment required with the 64-bit support, the IAM load library requires and is distributed as a PDSE with a block size of 32760. A normal PDS can be used, but modules will have to be copied to a PDSE should maintenance be necessary, and then copied back.

Resources

Additional information on the IAM product is available on request from Innovation. This includes the following:

- Concepts and Facility Guide
- Share Presentation (in Power Point format)
- Technical Newsletters
Summary of Modifications and Enhancements

IAM Version 9.2 provides a high performance alternative to VSAM for application programs that execute under IBM's z/OS operating system on z-Series processors. Enhancements in IAM Version 9.2 include IAM base product enhancements, enhancements for IAM/AIX, and enhancements for IAM/RLS and IAM/PLEX. The details of these product enhancements are described below.

z/HPF I/O

IAM Version 9.2 will utilize z/HPF I/O architecture I/O instructions on IAM Enhanced format files when the devices and software levels have the functionality necessary for using the architecture with EXCP level I/O processing. The specific IBM requirements are as follows:

- **Software:** z/OS 1.12 and z/OS 1.13 with APAR OA38185, or z/OS 2.1 and higher
- **Processors:** z196 GA2, z114, or zEC12 processor and above
- **Storage:**
  - IBM: DS8700 or DS8800 at 6.2 level and above
  - HDS: Hitachi Virtual Storage Platform G1000 at microcode level 80-02-01 or higher is appropriate for IAM zHPF support
  - EMC: Customers should contact their local EMC representative regarding information on an appropriate VMAX microcode level for IAM zHPF support.

IAM's use of z/HPF architecture I/O is automatic when all of the requirements are met. There are a couple of circumstances when z/HPF will not be used which are when writing out the DASD EOF block, and when the file extend process needs to format out additional extended area index blocks. The IAMINFO reports and SMF records include new statistics that indicate the number of I/O's that were done using z/HPF and those done with the standard CCW - ECKD channel programs.

The primary benefit of z/HPF architecture I/O is increased channel capacity. In testing done at Innovation's data center, this is best seen by the reduction in channel connect time. Our tests have seen an average of 26% reduction in connect time when utilizing z/HPF I/O architecture. A further possible benefit may be some reduction in elapsed time. Our tests have seen about a 4% reduction in elapsed time when using z/HPF. Because I/O time is only one of many factors in elapsed time, a job may not necessarily realize an elapsed time savings.

If an I/O error occurs with a z/HPF I/O, IAM will automatically retry the I/O with standard CCW channel program and temporarily suspend use of z/HPF on that volume for the dataset on which the I/O error occurred.

64-Bit Virtual Storage I/O Buffers

IAM Version 9.2 can optionally use 64-bit virtual storage for I/O buffers, without any need to change application programs. IAM is able to do this because IAM does not ever return a pointer to the I/O buffers it is using. Even when locate mode I/O is done, IAM will get a separate storage area and move the record into that area that is in storage that can be addressed by the application program. The advantage of using 64-bit virtual storage for buffers are that it will help reduce the demands on 31-bit addressable storage, and may enable large CICS, IAM/RLS, IAM/PLEX, or other long running address spaces to have more files open concurrently.
In our testing with 64-bit virtual buffers, it was noticed that there was an increase in CPU time used, particularly SRB time. With the tools we have available, the increase appeared to be primarily in page fix processing for 64-bit virtual storage. Because of the potential impact on CPU time, IAM does not default to using 64-bit virtual storage for buffers. However, users can set it as the default through the IAM Global Options, or use IAM overrides for the address space(s) where such buffering would be of benefit to eliminate virtual storage constraints.

I/O errors have occurred where the information provided in the IAMW37 I/O error message was insufficient to determine the real cause of the error. To help correct that diagnostic information, IAM will now display in the IAMW37 the full 32 bytes of sense information available to provide additional information in the rare circumstances that an I/O error does occur.

Users have requested some changes to our messages to facilitate automated operations responses to various messages that may occur while processing IAM files. IAM Version 9.2 will now utilize the multi-line WTO function of z/OS so that the dataset that the message is for can be more easily determined. In general, the IAM messages will have the dataset name in the last line of the messages issued. Most of the messages will only have two lines, but a few, such as an IAMW37 I/O error message with full sense information will require three lines. Additionally, the IAMW22 message which had four different possibilities for recommending an IAM file be reorganized has been changed to one of four separate message numbers which are IAMW21, IAMW22, IAMW91, and IAMW92.

Due to the 70 character per line limit imposed when using the multi-line format of the WTO service, many IAM messages have had there text revised to fit in the reduced amount of space. The changes were done with the intention of keep the messages with as obvious of a meaning as possible with the reduction in the amount of text for many of the messages.

IAM/RLS and IAM/PLEX

IAM Version 9.2 has an additional exit for CICS that along with other internal coding changes will enable IAM/RLS and IAM/PLEX to handle most of the VSAM RLS only file control options / operands, including UNCOMMITTED, CONSISTENT, REPEATABLE, and NOSUSPEND. CICS application programs that use those functions can now be used without change with IAM/RLS or IAM/PLEX with the full functional support.

IAMSMFVS Report Enhancements

IAMSMFVS has been enhanced to use 64-bit numbers for various statistics such as EXCP counts may exceed a 32-bit value. Additionally support for SMF interval records is now included which enables better reporting for datasets being used by long running address spaces, such as CICS or IAM/RLS without needing the step termination record. One other new feature is the ability for IAMSMFVS to produce a CSV (comma separated value) for datasets in the VSAM EXCP REPORT to facilitate easy analysis with a spread sheet.
Global Options

The following are new Global Options in IAM Version 9.2:

- **BUF64**  
  When enabled indicates that 64-bit virtual storage can be used for the I/O buffers for normal file access and update processing. The default is disabled, meaning that IAM will not use 64-bit virtual buffering.

- **CRBUFOPT** =  
  Has a new optional value of **64BIT** that indicates the IAM file load process is to use two megabytes of 64-bit virtual storage for the I/O buffers. The default value remains the same as prior versions at **MCYL**.

- **HPF**  
  When enabled Indicates that IAM will use the z/HPF channel architecture I/O instructions when the device, processor and z/OS level all have the support for that function. Default is ENABLED. When disabled, it will mean that the z/HPF I/O will not be utilized.

- **IAMW22LIM** =  
  Specifies the overflow index storage utilization level that will trigger the IAMW22 message indicating that a reorganization is recommended. Values are specified in megabytes, which defaults to 64, Valid values range from 16 to 1024.

- **NOMEMLIMIT**  
  When enabled will request IAM be permitted to exceed the MEMLIMIT for 64-bit virtual storage for the index subject to an IAM internal maximum of 64 gigabytes.

The default value for **DATASPACE** has been increased to **2048** due to some users having insufficient data space storage available for loading IAM datasets with a very large uncompressed index structure. Other Global Option values remain unchanged from the Version 9.1 defaults.

IAM Overrides

A change has been made for override processing for CICS, IAM/RLS, and IAM/PLEX. When running in any of those address spaces, the IAM override process will automatically reread the override statements whenever an IAM file is opened. This enables the overrides to be revised without requiring a re-cycle of those address spaces.

The following are the new or changed overrides for IAM Version 9.2:

- **BUF64 = YES | NO**  
  is a new ACCESS override indicating whether or not IAM is to use 64-bit addressable virtual storage for the I/O buffers. The default is from the Global Options Table, which as shipped is set to NO. There are no application program changes required to use this capability, as IAM will always move the data records requested into an area of storage that is addressable by the application.

- **CRBUFOPT = 64BIT**  
  is a new operand for the CREATE override of CRBUFOPT. Setting it to 64BIT enables IAM’s file load processing to use 64-bit addressable virtual storage during the file load. The default is taken from the Global Options which as distributed is set to MCYL, meaning 31-bit addressable storage is used.

- **HPF = YES | NO**  
  This is both a CREATE and ACCESS override that will enable or disable use of z/HPF architecture I/O. The override is provided in case there are any problems or issues using the HPF I/O capabilities. The default is from the Global Options which is set to YES as the product is shipped.

- **NOMEMLIMIT**  
  is a new ACCESS override that when specified will allow IAM to exceed the MEMLIMIT if necessary for storage for the index.
IAM Version 9.1

IAM Version 9.1 provides a high performance alternative to VSAM for application programs that execute under IBM’s z/OS operating system on z/Series processors. Enhancements in IAM Version 9.1 include IAM base product enhancements, enhancements for IAM/AIX, and enhancements for IAM/RLS and IAM/PLEX. The details of these product enhancements are described below.

Uncaptured UCB Support

IAM Version 9.1 when running under z/OS 1.12 or higher will now be able to utilize above the line UCB’s without being captured. The UCB’s will no longer need to be moved below the line while the data set is opened, helping to reduce the below the line storage requirements.

Support for VSAM Editors

When attempting certain functions under ISPF, such as browsing or editing IAM files, ISPF will now always invoke any installation specified VSAM software package such as File-Aid, File-Manager, Ditto, or other software.

New IAM Global Options

There are a few new Global Options in Version 9.1. The first two area controlled by the ENABLE and DISABLE keywords, which are EAV and LARGE. Both of these default to disabled. When the EAV option is enabled it will have all IAM files that are defined after the enabling to utilize the extended areas on EAV volumes, in effect setting IAM default to EATTR(OPT). When LARGE is enabled it will cause all newly defined IAM files to default to DSNTYPE=LARGE for Large Sequential Format dataset. Additionally, there is the CICSTXID Global Option that is used to set the name of the CICS transaction that will be invoked for the automatic reconnect when an IAM/PLEX or IAM/RLS address space returns to operation.

Automatic Recatalog for Renamed AIX Components

When IAM determines that a name of a component in an AIX sphere has been changed due to a copy process or rename, if IAM determines that there is an identical name change in the other related components, IAM will now automatically do the define recatalog that previously would have to be done manually.

Showcat Enhancement

IAM Version 9.1 enhances the support of the SHOWCAT service when used for IAM files to support the RETURN=NAME operand. Users can now obtain names instead of the old catalog CI numbers that are effectively tokens. This eliminates the need for the system and IAM to maintain internal tables in virtual storage to map the token to the catalog entry.

Persistent Record Locks

IAM Version 9.1 enables IAM/PLEX and IAM/RLS address spaces to re-establish record locks for recoverable files should the address space terminate, either on request or due to a failure. This is done through utilization of a new record lock journal file, and the data from the system logger or journals. This function enhances data integrity for files utilizing IAM/PLEX or IAM/RLS when an outage occurs. Use of this function is optional.

CICS Automatic File Disconnect and Reconnect

Enhancements to the IAM/PLEX and IAM/RLS support for CICS in the event of an IAM/PLEX or IAM/RLS address space termination. When it is detected that such an address space is no longer processing, all of the IAM files being processed through that IAM/PLEX or IAM/RLS address space will be closed. A list will be kept of the files that were being used so that they can be re-opened when the address space is active again. When the files are re-opened, any transaction back outs that had been deferred due to unavailability of the IAM files (shunted units of work) will be retried.
**Automatically Determine File Recoverability for CICS**
IAM will now determine automatically whether an IAM file being opened under IAM/PLEX or IAM/RLS by CICS is a recoverable file and manage the record locks as appropriate to that status. This eliminates the need to set the JRNAD=None override for files that are not recoverable under CICS.

**HiperDispatch Enhancements**
Performance enhancements to IAM/PLEX processing for z/OS systems that are utilizing the HiperDispatch mode. This includes reduction in local lock utilization, use of high performance Pause and Release function in place of Wait and Post, and changes to I/O task management and dispatching to minimize the need for the z/OS dispatcher to handle the I/O requests. This change reduces both SRB and TCB CPU times used by IAM/RLS and IAM/PLEX.

**New Journal Record**
A new journal switch record will be written when switching to a journal file to insure that at least one record has been written out to the new journal. This will prevent an unexpected reuse of the prior journal data set should the IAM/PLEX or IAM/RLS region be recycled after the switch but before any new records had been written out to it.

**Replication Logging**
New in IAM Version 9.1 is a replication logging function for IAM/PLEX and IAM/RLS. Referred to as Alternate Logging, this function will write out to a System Logger stream log records that are compatible with CICS format journal records intended to be used by products such as the RRDF™ Remote Recovery Data Facility from E-Net for data replication from CICS and batch jobs that are updating IAM datasets under either IAM/PLEX or IAM/RLS.

**CICS/TS V5.1 Support**
IAM Version 9.1 is the minimum level of IAM required for use with CICS/TS Version 5.1.

**IAM Version 9.0**
IAM Version 9.0 provides a high performance alternative to VSAM for application programs that execute under IBM's z/OS operating system on z/Series processors. Enhancements in IAM Version 9.0 include support for EAV super large volumes, increased utilization of 64-bit virtual storage to provide storage constraint relief, and enhancements to IAM Prime Related Overflow feature that can reduce the frequency of file reorganizations for datasets with heavy insert activity. The details of these and other product enhancements are described below.

**EAV Support**
IAM Version 9.0 supports IAM files on Extended Addressable Volumes (EAV). This function requires z/OS 1.11 or higher. For z/OS 1.11 requires that IAM files must be DFSMS Extended Format to reside on an EAV. For z/OS 1.12 and above the IAM datasets on EAV volumes can be Large Format Sequential or Basic Format Sequential. EAV eligibility is set by defining the file with a DFSMS Data Class that specifies the EATTR(OPT) attribute, or is defined with the IDCAMS DEFINE parameter EATTR(OPT). IAM files on EAV's cannot be accessed with prior versions of IAM.

**Prime Related Overflow**
Prime Related Overflow (PRO) is an alternative overflow structure within IAM files. The PRO overflow structure is designed and intended to be used for IAM files that have very high insert activity. For such files PRO provides the benefits of reduced virtual storage usage for the index structure and will reduce the frequency of reorganizations. The PRO support has been revised and enhanced in Version 9.0. PRO files created with Version 9.0 are not downward compatible with prior IAM releases.
64-bit Virtual Storage
IAM Version 9.0 can use z/OS 64-bit addressable virtual storage to hold the index for opened IAM files. All of the code involved in the processing of the index for IAM files has been converted to utilize the z/Architecture 64-bit instructions. Use of this function provides virtual storage constraint relief and eliminates the two gigabyte limitation for the indexes to IAM files. Depending on installation default values, users may need to use the MEMLIMIT JCL parameter on job steps that they desire to use this capability.

XTIOT & 31-bit DSAB Support
IAM Version 9.0 supports IAM files existing in the XTIOT (above the line TIOT area) and 31-bit DSAB control blocks. This support will allow users to allocate more datasets to their CICS, IAM/RLS, and IAM/PLEX address spaces. The XTIOT support currently excludes IAM files that are DFSMS Extended Format. This support requires that only IAM Version 9.0 or higher is active on a system, otherwise code from prior version will prevent the use of the XTIOT and 31-bit DSAB’s for IAM files.

Alter Capability
Through the DEFINE RECATALOG process and the applicable IAM CREATE overrides, various attributes of IAM files can now be altered. The information that can be changed are values for buffer space, journaling, maximum buffers, minimum buffers, the RLSID, and the share options.
Global Option Changes

Some of the default values of the IAM Global Options have changed in Version 9.0 to generally recommended values. Two of the Global Options are new for Version 9.0, as indicated by an asterisk next to the option. For the new options, the Version 8.1 value shown in parenthesis represents the effective value. The table below indicates the Global Option, the new default value for Version 9.0, and the default value in Version 8.1:

<table>
<thead>
<tr>
<th>Global Option</th>
<th>Version 9.0</th>
<th>Version 8.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BELOWPOOL</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>DATASPACE</td>
<td>1024</td>
<td>256</td>
</tr>
<tr>
<td>ESDSLOCK*</td>
<td>ENABLED</td>
<td>(enabled)</td>
</tr>
<tr>
<td>INDEXSPACE</td>
<td>64BIT</td>
<td>CICS</td>
</tr>
<tr>
<td>LONGKEYS</td>
<td>33</td>
<td>128</td>
</tr>
<tr>
<td>XESDS*</td>
<td>DISABLED</td>
<td>(disabled)</td>
</tr>
</tbody>
</table>

Figure 1-1: Table 1: New and Changed Global Options

IAM Load Library

Due to quad-word alignment with the 64-bit support, the IAM load library requires and is distributed with a block size of 32760. In order to apply maintenance to some of load modules, it also must be a PDSE. A normal PDS can be used, but modules will have to be copied to a PDSE should maintenance be necessary, and then copied back.

Error Message Enhancements

To aid with problem diagnosis, most error messages will now have a second line containing the affected dataset name. Additionally IAM now supports the ACB message area, passing back open error messages if an open fails for an IAM file to the application program, in addition to writing the message to the job log.

IAM/RLS

Based on user input, a number of modifications have been made to IAM/RLS that are summarized as follows:

There will be one dataset name table that is applicable to all of the IAM/PLEX and IAM/RLS address spaces within a Sysplex. This provides one central location to control the use of IAM record level sharing capabilities. To facilitate a single table, a new parameter of RLSID=nnnn is added on the SELECT and EXCLUDE statements to indicate which IAM/RLS or IAM/PLEX address space can select or exclude the datasets from processing. Additionally, an enhanced dataset name masking capability is provided for the specification of dataset names on the SELECT and EXCLUDE statements. This will help to reduce the size of the table as it exists both in source format and in internal format that is kept in virtual storage.

64-bit addressable virtual storage can now be used to contain the record locks within each IAM/RLS and IAM/PLEX address space. This capability will increase the number of record locks that can be held, and reduce contention for 31-bit addressable storage. This is enabled by specification of the LOCKMEM=64BIT IAM/RLS and IAM/PLEX startup parameter. Use of this capability will generally require users to specify the MEMLIMIT keyword on the EXEC card for the IAM/RLS or IAM/PLEX JCL procedures.

An automatic sync point process is now available for batch jobs. This will be controlled by a new ACCESS override of AUTOSYNCPPOINT=nnnnn which specifies that a sync point will be taken after the specified number of lock obtaining requests have been issued.
RLSINFO, a new statistical report, can be automatically produced on a periodic basis for both IAM/RLS and IAM/PLEX address spaces. Similar to the IAMINFO report, these reports provide information on the activity that has occurred within the particular IAM/RLS or IAM/PLEX address space. This report is enabled by the STATS=ON startup parameter, and the specification of an RLSINFO DD card in the proc used to start IAM/RLS or IAM/PLEX.

Various error messages for a dataset, particularly for open failures, are passed back to the program that issued the open in the ACB message area, and are also displayed on the job log. This is to help diagnose problems reducing the need to examine the RLSLOGDD or the job log of the IAM address space to find error messages.

A number of internal processing changes were made that are intended to help performance for IAM/RLS and IAM/PLEX processing and to reduce some of the CPU time used. Included in the changes is a reduction in use of the local lock for serialization. This has included the reducing or elimination of use of various system services, along with implementation of different serialization techniques to reduce the resource contention.

IAM Version 9.0 presents IAM/PLEX, a z/OS SYSPLEX record level sharing capability for VSAM applications that is available as an additional cost feature of the IAM product. This feature enables VSAM applications to execute concurrently and update IAM files on multiple LPAR’s / systems that are part of a z/OS Sysplex. The applications sharing these files can be any mix of CICS/TS transactions, batch programs, or TSO applications. The IAM/PLEX approach to Sysplex record level sharing of datasets is intended to provide users with record level sharing capabilities without the implementation complexity and application changes that are typically necessary for implementing VSAM record level sharing. The IAM/PLEX function utilizes the z/OS Sysplex XCF services, the z/OS System Logger facilities, and the IAM/RLS framework to enable the sharing of datasets at the record level across multiple LPAR’s.

Details on IAM/PLEX are in Section 2.20 and Section 25 of the IAM Manual.
01.03 Organization of This Manual

Format of the Manual

The IAM manual is divided into sections, by a major general topic, identified by an integer such as 10 or 20. Sections are divided into subsections by subtopic, with identifiers such as 10.20 or 30.02. Within subsections, subtitles may appear in the left margins to help you quickly find topics of interest. Each page displays its subsection title in the top center of the page, and the subsection number at the top right corner. Within each section, pages are numbered with the section number and a sequential page number within the section, e.g., page 80-12.

As much as possible the section numbers used in prior versions of the manual remain the same in the IAM Version 9.0 manual. The main change in the organization of the IAM Version 9.0 manual is that section 00 from prior versions has now been merged into section 01 and a new section 25 has been added for the IAM/PLEX feature. The benchmark data that was in Section 00.02 of the IAM V8.1 manual has been removed from the manual, and will be available in an updated version of the IAM White Paper.

Contents of the Manual

The IAM manual contains documentation and instructions for utilizing the IAM product including the optional features IAM/AIX and IAM/PLEX. Therefore portions of the manual may not be relevant if you are not licensed for the feature being discussed. Documentation regarding the older IAM compatible format files has been removed from the IAM Version 9.2 Manual, but remains available in prior manual versions, such as in the IAM Version 8.1 Manual. IAM Version 9.2 does still support compatible format files however it has been functionally stabilized for over ten years, and does not provide access to all the features and functions of IAM.

All users are encouraged to utilize the standard IAM Enhanced file format. Users who have the FDREPORT product can use that to find all of their compatible format files. Below is an example of the JCL and control statements to do so:

```
//IAMFILES EXEC PGM=FDREPORT,REGION=0M
//SYSPRINT DD   SYSOUT=* 
//ABRMAP   DD   SYSOUT=* 
//SYSIN    DD   *
XSELECT DSORG=IAM
XEXCLUDE IAMINDIC=ENHANCED
REPORT FIELDS=(DSN,VOL,SIZE,NOEXTENT,IAMINDIC,DATES)
TITLE LINE='IAM COMPATIBLE FORMAT FILES'
SUMMARY FIELDS=(DSN)
PRINT ENABLE=IAM,SORT=COMBINE,
       SORTALLOC=(SORTWORK,CYL),
       ONLINE,SUM=YES
/*
```

Figure 1-2: Figure 1: Example JCL to use FDREPORT to find Compatible Format files (EX0103A)
### Improving the Manual

We have tried to make this manual as complete, precise and error free as possible. However, in spite of our best efforts, errors and incomplete explanations may have crept in. Should you encounter any of these, we would appreciate your corrective criticism. It is primarily through your feedback that we can improve this manual. There is a Reader's Comment Form at the back of the manual; you may mail it, fax it, or send us e-mail.

### Examples

All examples and Job Control Language statements shown in this manual are for ILLUSTRATIVE PURPOSES ONLY! You must modify them to meet the JCL requirements of your installation.

The JCL examples in this manual are also included in a JCL library on the distribution media. If you load this JCL library to disk, you can reduce the time required to setup jobs for various IAM functions by using the JCL library members as a starting place.
Control Statement Format

01.04 Control Statement Format

General
The IAM control statements consist of 80-character logical records. The general format of these records is:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>OPERANDS</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

WHERE:

Columns 1 to 71 - contain the command, operands, and comments fields, except when continued to subsequent logical records.

Column 72 - must be blank if the command or operands extend to column 71.

Columns 73 to 80 - not used by IAM. We suggest you use them as an identification or sequence field.

Command Field
The command field identifies the control statement and consists of a one or more character command word. It may appear anywhere within columns 1 to 71 and can only be preceded by blanks. The command word must appear in its entirety within columns 1 to 71; it may not be continued.

Operand Field
The operand field, if present, follows the command field and is separated from it by at least one blank. The operand field consists of one or more keywords and/or positional parameters, separated by commas. It may not contain embedded blanks except within quoted strings. Operand fields may be continued onto subsequent logical records. If operands are to be used with a command, at least one operand must be on the logical record containing the command.

Comments Field
The comments field, if present, follows the operand field(s) and is separated by one or more blanks. It may contain any information deemed helpful by the person who codes the control statement. Comments fields may not be continued, i.e., they must end at or before column 71. Comments are not permitted on a control statement that allows operands but on which no operands have been specified.

Coding an Operand Field
An operand field consists of one or more positional or keyword parameters separated by commas.

Positional Parameters
Positional parameters must be coded in a specific order relative to one another. This means that the variable data you substitute for positional parameter 1 must precede the variable data for positional parameter 2, and so on. The absence of variable data to be substituted for a positional parameter is indicated by coding a comma in its place. However, you may omit the commas when the absent parameter is the last one, or if all following positional parameters are absent.
Keyword Parameters
Keyword parameters are position independent, and consist of either a keyword alone or a keyword followed by an equal sign (=) followed by user-specified variable information. When both positional and keyword parameters are to be coded in an operand, keyword parameters must precede positional parameters.

Sub-parameters
Both positional and keyword parameters may consist of a list of subparameters. Such a list is composed of positional parameters that follow the usual rules for that type. A subparameter list must be enclosed within parentheses, unless the list reduces to a single subparameter, in which case the parentheses may be omitted.

Quoted Strings
When the variable data you specify for a parameter contains certain special characters, defined below, you must enclose the data with apostrophes. This is called a 'quoted string'. Within a quoted string, all characters, including blanks, may appear; if an apostrophe is to be part of the string, it must be coded as two apostrophes.

NOTE: Special characters are parentheses, commas, equal signs, apostrophes, and blanks.

Continuing an Operand Field
When the total length of an operand field exceeds the available columns in a logical record, it must be continued onto one or more following logical records. To continue an operand, interrupt the operand field after a complete parameter or subparameter, including the following comma. Do this at or before column 71. Leave the next column blank.

Continue the operand field starting anywhere between columns 1-71 on the following logical record.

Any number of logical record continuations may be present.

Example Continued Statement

```
REPORT MAXDSNS=6000,
DETAIL,
GROUPNAMES=(CICS,ABC),
MAXREPORTS=5000
```

Example of a Continued Operand Field, With Comments

Figure 1-4: Figure 3: Example of a Continued Control Statement

Comment Statement
If you really have a lot to say, you may code a comment statement, which are control statements with nothing but remarks contained within columns 2 to 71. Comment statements are identified by an asterisk (*) in column 1. They may appear anywhere within a group of control statements, even between continued logical records.
01.05 Documentation Notation Format

Notation

The following notation is used in this manual to define control statement formats:

**Uppercase letters and words** must be coded exactly as shown in a format description.

**Lowercase letters and words** represent variables for which you must substitute specific information.

**Brackets** are never coded. They indicate that the enclosed item is optional, and you can code one or none of the items.

For Example: `[MAXDSNS=nnnn ]`

**An ellipsis** . . . (3 consecutive periods) is never coded. It indicates that the preceding item can be coded more than once.

For Example: `DSN=(dsname,dsname . . . )`

**An underscore** ______ is never coded. It indicates that the underscored item is the abbreviation for the value.

For Example: `DATACOMPRESS= YES | NO`

**A vertical bar** | is never coded. It indicates that there is a choice between the values specified.
### Vendor Products That Have Converted VSAM Files to IAM

This is a sample of some of the Vendors’ products that Customers have converted VSAM files to use IAM:

<table>
<thead>
<tr>
<th>VENDOR</th>
<th>PRODUCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accero (CYBORG)</td>
<td>Payroll</td>
</tr>
<tr>
<td>American Software</td>
<td>DRP</td>
</tr>
<tr>
<td>ASG (Mobius)</td>
<td>ViewDirect</td>
</tr>
<tr>
<td>Atos Origin</td>
<td>CARDLINK</td>
</tr>
<tr>
<td>CGI</td>
<td>CGI Advantage</td>
</tr>
<tr>
<td>CSC</td>
<td>Hogan, Cyberlife, Capsil</td>
</tr>
<tr>
<td>CSI</td>
<td>Bank Trade</td>
</tr>
<tr>
<td>Fidelity National</td>
<td>Systematics Banking Appl, BASE2000, CSF</td>
</tr>
<tr>
<td>First Data</td>
<td>VisionPLUS</td>
</tr>
<tr>
<td>Fiserv</td>
<td>Pep Plus, Mortgage-Serv</td>
</tr>
<tr>
<td>Infor (GEAC)</td>
<td>GL, AR, MSA, Millennium, Walker, Infopoint Deposits</td>
</tr>
<tr>
<td>JDA</td>
<td>Compass Contract</td>
</tr>
<tr>
<td>LRS</td>
<td>PageCenter</td>
</tr>
<tr>
<td>LSI</td>
<td>Popims</td>
</tr>
<tr>
<td>Mackinney Software</td>
<td>JES Queue Print</td>
</tr>
<tr>
<td>McKesson (HBO)</td>
<td>HealthQuest</td>
</tr>
<tr>
<td>Pitney Bowes (Group 1)</td>
<td>Finalist, Code 1, Mailstream Plus</td>
</tr>
<tr>
<td>Retalix (NCR)</td>
<td>Biceps, ABS</td>
</tr>
<tr>
<td>SEA</td>
<td>TRMS, SAVRS</td>
</tr>
<tr>
<td>Serena</td>
<td>Changeman ZMF</td>
</tr>
<tr>
<td>Shaw Systems</td>
<td>Collections</td>
</tr>
<tr>
<td>Siemens Medical Systems</td>
<td>Invision, Signature</td>
</tr>
<tr>
<td>Sigma</td>
<td>SAM (Student Aid Management)</td>
</tr>
<tr>
<td>SunGuard (SCT)</td>
<td>SIS+, HRS, FRS</td>
</tr>
<tr>
<td>Empower Software (Tesseract)</td>
<td>Payroll, HR, Benefits</td>
</tr>
<tr>
<td>TriZetto</td>
<td>ClaimFacts, GroupFacts</td>
</tr>
<tr>
<td>VIPS</td>
<td>Medicare Part B, SuperOP</td>
</tr>
</tbody>
</table>
IAM works with many system management software products that are used with VSAM today. Below is a list of products known to work with IAM. In many circumstances, these vendors have made enhancements to their products so that they can be used with IAM files. This is by no means a complete list of all the products that do work with IAM files.

- **DASD MANAGEMENT SOFTWARE:**
  - FDR/ABR (INNOVATION DATA PROCESSING)
  - FDRREPORT (INNOVATION DATA PROCESSING)
  - FDRREORG (INNOVATION DATA PROCESSING)
  - DF/SMS (IBM)
  - DF/HSM (IBM)
  - DF/DSS (IBM)
  - CA ALLOCATE (formerly VAM) (CA)
  - DMS/OS (CA)
  - POOLDASD (BOOLE & BABBAGE)
  - MAINVIEW SRM STOPX37/II (BMC)

- **JOURNALING AND RECOVERY:**
  - FILE SAVE (CA)
  - DRS (BMC)
  - AR/CTL (BMC)
  - CICS/VR (IBM)
  - RRDF - Remote Recovery Data Facility (E-Net)

- **MISCELLANEOUS PRODUCTS:**
  - FILE-AID (COMPUWARE)
  - ABEND-AID (COMPUWARE)
  - CICS (IBM)
  - CONNECT: DIRECT (IBM-Sterling)
  - ISPF (IBM)
  - NETWORK DATA MOVER (CA)
  - (NDM)
  - SELCOPY and SELCOPY/i (CBL Compute (Bridgend) Ltd)
  - SHRINK (CA)

- **PERFORMANCE MONITORS:**
  - OMEGAMON (TIVOLI)
  - THE MONITOR TMON (ASG)
  - STROBE (COMPUWARE)

- **PROGRAMMING LANGUAGES:**
  - VS/COBOL (IBM)
  - COBOL II (IBM)
  - FORTRAN (IBM)
  - PL/1 (IBM)
  - BAL (IBM)
  - CA/OPTIMIZER (CA)
  - SAS (SAS INSTITUTE)
• **SECURITY PRODUCTS:**
  - RACF (IBM)
  - ACF/2 (CA)
  - TOPSECRET (CA)

• **SHARING PACKAGES:**
  - IAM/PLEX (IDP)
  - SHARE OPTION 5 (CA)
  - SYSB (H & W)

• **SORT PRODUCTS:**
  - SYNCSORT (SYNCSORT)
  - DF/SORT (IBM)
  - CA/SORT (CA)
2 IAM CONCEPTS AND FACILITIES

IAM Overview

IAM Basics
IAM is a high performance indexed access method for z/OS that offers numerous advantages over the IBM provided VSAM access method. IAM files exist on DASD in a different file structure than VSAM datasets, with IAM providing a VSAM compatible API (Application Programming Interface) for KSDS and ESDS file types. IAM/AIX available as an additional cost feature provides support for alternate indexes and RRDS file types. Existing as non-VSAM data sets provides IAM with capabilities to eliminate the 4.3-gigabyte file size restriction in VSAM for files that are not DFSMS Extended Format with Extended Addressability, and to choose a block size that will optimize space utilization on each of the different type of DASD devices and architectures available. Along with IAM's unique file structure, described in a subsequent section and IAM's Data Compression feature, user data stored in an IAM file typically requires substantially less DASD space then when stored in a VSAM cluster.

The basic principle of IAM is to intelligently utilize virtual storage to reduce the need to perform physical I/O to look up and retrieve data. This is accomplished by using the sophisticated IAM Real Time Tuning concept to manage buffers, and by keeping the index for files in virtual storage while they are being processed. IAM requires at most one I/O to retrieve any explicitly identified record within the IAM data set. All of the index I/O's and index buffers used by VSAM are eliminated.

Dynamic File Expansion
IAM files can acquire additional DASD extents as necessary subject to the type of dataset that is being used. IAM supports basic non-VSAM file structure, the large sequential dataset structure, the SMS sequential extended format, and SMS sequential extended format on EAV volumes. In all cases files are limited to a maximum of 59 volumes.

<table>
<thead>
<tr>
<th>DSNTYPE</th>
<th>Extents per volume</th>
<th>MAX Extents</th>
<th>MAX Space per Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASIC</td>
<td>16</td>
<td>255</td>
<td>64K Tracks</td>
</tr>
<tr>
<td>LARGE</td>
<td>16</td>
<td>944</td>
<td>64K Cylinders</td>
</tr>
<tr>
<td>LARGE FORMAT + EAV (z/OS 1.12 and above)</td>
<td>16</td>
<td>944</td>
<td>&gt;64K Cylinders</td>
</tr>
<tr>
<td>EXTENDED FORMAT</td>
<td>123</td>
<td>7,257</td>
<td>64K Cylinders</td>
</tr>
<tr>
<td>EXTENDED FORMAT + EAV (z/OS 1.11 and above)</td>
<td>123</td>
<td>7,257</td>
<td>&gt;64K Cylinders</td>
</tr>
</tbody>
</table>

Figure 2-1: Table 2: DSN Type Specifications
Maximum File Size

The maximum size of an IAM file within the internal file architecture is 128 terabytes. Given the combination of the IAM software implementation and the 3390 track architecture, assuming a ½ track blocking factor yields a maximum of 54 terabytes.

In real world terms, the maximum size is subject to the 3390 track architecture and the limits indicated in the table shown above. The current major factor beyond actual volume size is the limit of 59 volumes per dataset. Given that consideration, the maximum size of an IAM file on all 3390-54 size volumes, assuming either large format or extended format, is approximately 3,186 gigabytes of compressed data. On all EAV volumes with the current maximum of approximately 223 gigabytes per EAV volume, that yields a maximum IAM file size of 12.8 terabytes. For a basic non-VSAM format file, the limit is 201 gigabytes of compressed user data due to the 64K tracks per volume maximum.

The above size limitation applies to IAM KSDS file types, RRDS file types, and to IAM ESDS file types defined with the XESDS attribute. For applications that require a true 4-byte VSAM RBA value for an ESDS file the size limitation is 4 gigabytes of user data.

VSAM Transparency

IAM's system level VSAM Interface provides application program transparency. IAM allows an unaltered application program executing under z/OS to access IAM files in place of VSAM KSDS, ESDS, RRDS, and AIX files. IAM can be used in conjunction with the common programming languages COBOL, BAL (assembler), PL/1, RPG, C and any higher-level language products, which support keyed access to VSAM files.

IAM supports programs executing in AMODE(31) and VSAM control blocks (ex: ACB, RPL) residing above the 16MB line.

IAM supports the full range of VSAM file access commands GET, PUT, INSERT, GETPREV, ERASE, POINT, etc. and the file status commands SHOWCB, TESTCB, GENCB, and the VSAM catalog lookup macro, SHOWCAT.

IAM supports the following functions of IDCAMS, as they relate to VSAM KSDS, ESDS, RRDS and AIX file processing: BLDINDEX, DEFINE, DELETE, LISTCAT, PRINT, REPRO and VERIFY. DELETE, PRINT, REPRO and VERIFY provide the same services for IAM files as they would VSAM clusters.

IDCAMS DEFINE will create an IAM file whenever the OWNER($IAM) parameter is specified, 'IAM' is placed somewhere in the data set name, or $IAM is part of the Data Class or Storage Class name.

LISTCAT ALL displays IAM files as non-VSAM in its standard SYSPRINT report. LISTCAT ALL also displays the file's IAM characteristics in an IAMPRINT DD report, which will be dynamically allocated if necessary.

IAM has full support for the SMS environment. This support includes recognizing and using the SMS classes for allocation, honoring file attributes specified in the Data Class, and support for JCL allocation of IAM files; including temporary data set support. Users can also use an SMS Data Class name or Storage Class name that contains the literal $IAM to define a file as IAM. When using the SMS JCL allocation feature specifying $IAM in the data set name, in the Storage Class name or in the Data Class name on the DD Statement results in an IAM file being allocated.
Activating the IAM VSAM interface is a simple procedure. While evaluating IAM, you can activate and deactivate IAM at any time. To start IAM all you need to do is submit the procedure, ‘VIFSTART’, supplied in the IAM Installation Control Library. Within a few seconds, IAM will be active in the system. Once testing has been completed and IAM is in production, you can activate IAM automatically each time the system is IPL’d.

Section 90.20 of this manual contains information on how to activate the IAM VSAM Interface.

Record Level Sharing
IAM provides a record level sharing capability that starts within the scope of a single address space, and is extended by IAM/RLS to multiple concurrent online regions and batch jobs within a z/OS image up to the IAM/PLEX feature that enables IAM datasets to be concurrently shared across multiple z/OS images within a Sysplex.

IAM’s basic processing incorporates a natural record level sharing within the scope of a single address space. IAM does not lock at the CI level, which can delay access to records within a CI even when they are not the object of an update process with LSR, but just happen to be in a CI that does contain a record being updated. This is a particular benefit for online systems, where transactions can be delayed by VSAM due to the CI lockout process. While many users and CICS itself have implemented strategies to minimize the CI lockout, it is still an issue with VSAM that doesn’t exist with IAM.

IAM/RLS
IAM/RLS is a feature included in the base IAM product that provides sharing of IAM datasets at the record level between multiple concurrently running applications within a single z/OS image (LPAR). This enables multiple online (CICS) regions to concurrently share IAM files for update with other CICS regions, batch jobs and TSO users. IAM/RLS includes journaling and recovery capabilities that can be of particular benefit for batch jobs, as CICS already provides recovery for transactions that works with IAM/RLS.

IAM/PLEX
IAM/PLEX is an optional additional cost feature to the IAM product that provides record level sharing of IAM datasets across multiple z/OS images (LPARS) that are within a Sysplex. IAM/PLEX is built on the framework of the IAM/RLS software with the objective of providing an easier mechanism to implement record level sharing at a lower cost than full VSAM/RLS implementation. IAM/PLEX includes all the functions of IAM/RLS including the journaling and recovery capabilities. The intention of the design of the IAM/PLEX is to provide high availability of data access. IAM/PLEX does not guarantee continuous availability or eliminate single points of failure.
IAM Alternate Index Support

02.10

IAM Alternate Index Support

IAM offers, at an additional cost, Alternate Index Support. With this optional feature IAM data sets (both KSDS and ESDS) can have associated alternate index data sets, which will also be IAM data sets. All of the features and benefits of IAM are available to base clusters and all of their related alternate indexes. There are no programming changes necessary to use the IAM Alternate Index support. No changes are required to CICS. Simply change the define of the base cluster to include either OWNER($IAM) or include $IAM in the base cluster name, or use a DATACLAS or STORCLAS with $IAM in the class name. Subsequent define steps for any related alternate index data sets or paths will be automatically converted to IAM, when the base cluster is an IAM data set. The alternate index support is designed to provide the same high level of compatibility with VSAM alternate indexes that IAM has been providing for KSDS and ESDS types of data sets. Some data set management utility jobs for backing up, restoring, and renaming data sets involved in a VSAM alternate index relationship will need to be revised.

The IAM Alternate Index

An alternate index provides an additional index to an indexed type of data set (KSDS), or an index to an entry sequence data set (ESDS). Users can define one or more alternate indexes to any base data set. The alternate index data set itself is an IAM KSDS data set that is indexed by the alternate key. The records in the alternate index contain some control information, the alternate key, and the primary key value for the record in the base data set with the corresponding alternate key. An alternate index is defined as containing UNIQUE keys when there is only one base record with any specific alternate key. Alternate indexes can also be defined as containing NONUNIQUE keys, where any particular alternate key can be contained in multiple base records. As with VSAM, the alternate index data set can be explicitly processed by programs without referencing the base cluster, or can be used to access the records in the base cluster. To use an alternate index data set to access the records in the base cluster, a PATH must be defined.

An additional attribute of alternate index data sets is whether they are upgradeable. When an alternate index is defined with the UPGRADE attribute, IAM will automatically update the alternate index either when updates are made to the base cluster, through the primary key or when accessed through an alternate key. Any alternate index defined with the NOUPGRADE attribute will not be automatically updated by IAM, and it is the application programs responsibility to ensure that the alternate index is updated in a manner to remain synchronized with the base data set.

The IAM Path

A Path provides the mechanism to access a base cluster through an alternate index. When a path is defined, one must provide the name of the path and the name of the related alternate index to be used to access the base cluster, or the base cluster to be processed whenever the path itself is referenced. The UPDATE or NOUPDATE attribute is specified for each path defined. When a path is defined with the NOUPDATE attribute, any upgradeable alternate indexes will not be updated automatically when this path is used to access a base cluster either directly or through an alternate index.

For VSAM data sets, a PATH is only a catalog entry that is quite similar to an ALIAS entry. With IAM, a PATH will become a one-track data set containing the name of the related alternate index or base cluster and the UPDATE or NOUPDATE attribute of the PATH. A single track data set was chosen instead of an ALIAS type of catalog entry because of concerns about there not being adequate data set management utility program support for ALIAS entries, which might result in the loss of the entry.
Using an IAM Alternate Index

The procedure for defining and using IAM data sets with alternate indexes is identical to the VSAM process. The only difference is to specify on the DEFINE that the base cluster is to be an IAM data set. This specification is typically done by either adding the OWNER($IAM) parameter to the define parameters, changing the name to include the $IAM literal, or using a DATACLAS or STORCLAS with $IAM in the class name. Most applications that are going to be using IAM for their alternate index already have the general procedures established. To convert to IAM, the only change needed is on the define step of the base cluster. An overview of the process for using IAM alternate indexes is described below. All of these steps are covered in detail in the User’s Guide (Section 10) of this manual.

- Defining the base cluster in Section 10.20.
- Loading the base cluster with data, in Section 10.30.
- Defining the alternate index(es) in Section 10.61.
- Building the alternate index(es) typically done with IDCAMS BLDINDEX command in Section 10.62.
- Defining paths to the alternate index(es) and if used, paths to the base cluster in Section 10.63.
- Using paths to access data in Section 10.64.
- Special Considerations with IAM Alternate Indexes in Section 10.65.

In general, there are no JCL changes required when using IAM Alternate Indexes and Paths. To access a base cluster through an alternate index, simply specify the name of the Path on the DD statement, just as is done with VSAM. When the Path is opened, IAM will dynamically allocate and open all of the required alternate indexes and the associated base cluster using the relation information stored within the IAM file structure. When the Path is closed, IAM will close and de-allocate all of the alternate indexes and the base cluster that were allocated during the open process.

Because the IAM alternate indexes and paths are non-VSAM data sets, they will not be automatically grouped together in a sphere with their associated base cluster, as is typically done by many data set management utility programs. Some changes to your data set management procedures may be necessary when using IAM Alternate Indexes. After renaming a component of an IAM sphere, or performing a restore or copy to a new name, a define recatalog may be necessary. IAM will attempt to automatically perform the define recatalog when an IAM AIX is opened when the identical name change was made to each of the components, that is base cluster, alternate index, and paths.
IAM/PLEX: IAM Sysplex Record Level Sharing

The IAM/PLEX feature provides record level sharing support of IAM datasets across multiple LPARs within a z/OS Sysplex. IAM/PLEX builds upon the IAM/RLS function by utilizing the XCF Sysplex services and the Sysplex Logger to provide a record level sharing capability for IAM files within a Sysplex. IAM’s implementation is an evolutionary step forward from the basic single system IAM/RLS that utilizes the IAM/RLS framework that has been in use for several years. The IAM/PLEX design is different from VSAM/RLS and is intended to enable users to take advantage of record level sharing without the complexity involved with setting up a VSAM/RLS environment. IAM/PLEX provides the following capabilities for VSAM applications:

- Concurrently share data in IAM files at the record level for update by batch and online applications from multiple LPARs and systems in a Sysplex. This capability provides improved availability of data on online systems by eliminating the need to close files while batch updates are being performed.
- Automatic record locking without application program changes. Batch jobs that do extensive update activity may need to utilize the IAM sync point process to avoid holding excessive record locks.
- Support of the CICS VSAM RLS parameters UNCOMMITTED, CONSISTENT, REPEATABLE, and NOSUSPEND. IAM/PLEX provides equivalent functionality for these parameters.
- Journaling capabilities that can be used for recovery and auditing purposes.
- Capabilities to back out updates for failed applications, including an automatic dynamic back out of updates by a batch job step that abends.
- Capabilities to recover datasets that may have experienced a media failure.
- Explicit or automatic batch sync point processing for batch jobs that do heavy update processing.

All of the logical and physical I/O processing for files handled by IAM/PLEX will be performed by an IAM/PLEX address space. IAM/PLEX operates with a concept that any particular dataset will be "owned" by one instance of an IAM/PLEX address space within the Sysplex. Each IAM/PLEX instance will be responsible for all the I/O, record locking, and journaling services for the datasets that it owns. I/O requests for IAM datasets handled by remote IAM/PLEX instance, such as an IAM/PLEX instance on another LPAR, will be sent to the owning IAM/PLEX instance by utilizing the z/OS Sysplex XCF signaling services. The performance of the remote I/O process is dependent on the configuration of the XCF communication links between the LPARs.

Datasets that are being processed by IAM/PLEX are assigned an RLSID, which is a four character identifier specified by the user. The RLSID specifies the named instance of IAM/PLEX that will be the owner that will process the I/O requests for the subject datasets. The RLSID is assigned to a dataset either by the IAM/PLEX dataset name table, or through the use of IAM CREATE or ACCESS overrides. Each IAM/PLEX instance is assigned an RLSID through startup parameters.
In addition to the RLSID, each IAM/PLEX instance is assigned to a group of IAM/PLEX instances referred to as the RLSGROUP. The IAM/PLEX RLSGROUP corresponds to the z/OS XCF group that will be used for communication within the group. An IAM/PLEX instance can communicate only with other IAM/PLEX instances that are in the same RLSGROUP. In effect then, each IAM dataset that is using IAM/PLEX services with an RLSID also becomes a part of the RLSGROUP that the owning IAM/PLEX instance is in. To access that dataset there needs to be an IAM/PLEX instance that is a member of that RLSGROUP on each LPAR requiring such access.

**IAM/PLEX Diagram**

In the above diagram there is one RLSGROUP named RLSPROD, which has two IAM/PLEX instances named RLS1 and RLS2. Each instance is running on a separate LPAR.
For example let's say that CICS-1 wants to read a record for update in IAM DSN=PROD.B which is owned by IAM/PLEX_RLS2. CICS-1 issues a standard read for update, which IAM processes and sends the request from IAM/PLEX_RLS1 to IAM/PLEX_RLS2 using XCF services. IAM/PLEX_RLS2 will obtain the record lock, retrieve the record, then send the record through XCF back to the CICS-1 address space through IAM/PLEX_RLS1. In this scenario, IAM/PLEX_RLS1 is known as the router, and IAM/PLEX_RLS2 is the target.
When configuring how to set up an IAM/PLEX, there are some basic concepts that form the foundation of how IAM/PLEX works. The most essential basic concepts are the following:

1. A job step, a CICS region, or any other address space can directly connect to only one IAM/PLEX instance. All of that user address space communication with other IAM/PLEX instances will be through that one IAM/PLEX instance. Therefore all of the IAM datasets that an address space accesses through IAM/PLEX must be owned by an IAM/PLEX instance that is within the same RLSGROUP.

2. Any particular dataset can only be owned by one IAM/PLEX instance. Therefore any user address space that needs to access that dataset through IAM/PLEX must be either directly connected or remotely connected through the same RLSGROUP.

3. An IAM/PLEX instance can belong to only one RLSGROUP.

4. There can be multiple RLSGROUPs within a Sysplex. Each RLSGROUP is completely independent of the other RLSGROUPs. Examples might be a production RLSGROUP, and a test RLSGROUP. Or there could be RLSGROUPs for different business units, where each group has its own set of unique datasets with no need to access datasets in different RLSGROUPs.

5. There can be multiple instances of IAM/PLEX address spaces on a single LPAR. When multiple IAM/PLEX instances that are members of the same RLSGROUP are active on the same LPAR only one instance will be eligible for direct connection by users on that LPAR. The other instances of the same RLSGROUP are treated as remote instances with services provided through XCF.

6. A shared Sysplex system logger is required if journaling is being used.

7. Persistent record lock support is available with the use of a Record Lock Journal dataset. If an IAM/PLEX instance is recycled, the record locks it held at the time it was shut down or failed will be reestablished by reading the Record Lock Journal dataset and the Sysplex system logger. Each instance of an IAM/PLEX that owns datasets will need its own Record Lock Journal to facilitate this process.

8. An automatic disconnect and reconnect function is available for CICS regions that are utilizing IAM/PLEX services. This will help prevent the need to recycle CICS regions, automatically re-open files that were being processed at time of failure when the IAM/PLEX instance is available and initiate any transaction backouts that may be necessary due to the failure.

9. Based on our testing configuration, while a Coupling Facility can be utilized for XCF communications, the performance is significantly better through the use of channel to channel (CTC) type links. Our configuration does not have a dedicated CF engine, so we do not have any data at this time for that type of configuration. Depending on the workload, multiple pairs of links between each LPAR involved may be necessary to achieve the desired performance. There are many components involved in IAM/PLEX performance, from our environment the XCF communication is the major component utilizing the most time.

Please also read the next section, 02.30 on the IAM Record Level Sharing for additional information on the features and capabilities of IAM/PLEX.

Full information on setting up and using IAM/PLEX is in Section 25 of the IAM Manual.
IAM Record Level Sharing

The IAM single system Record Level Sharing permits multiple jobs, CICS regions and TSO users to concurrently update IAM files, with full data integrity. Locking is performed at the record level, so the different jobs can update records within the same data block without any delay. With this capability built into IAM, customers can come even closer to achieving continuous availability of online transaction processing systems. IAM’s Record Level Sharing additionally provides for journaling file update activity, and the ability to either backout updates from failing batch jobs or TSO users, or to perform a forward recovery if an IAM file had to be restored from an earlier backup.

IAM/RLS is easy to use. The setup checklist for preparing for IAM/RLS includes the following:

1. Allocating an IAM/RLS parameter library, and choosing the startup parameters.
2. Allocating the IAM/RLS journal data sets, if journaling is going to be used, and setting up backup and empty procedures for the journal data sets.
3. Allocate a Record Lock Journal for each IAM/RLS address space for the persistent record lock capability.
4. Deciding on eligibility criteria for IAM/RLS usage and building the data set name table if it is going to be used. Set the desired option for RLS processing in the IAM Global Options Table.
5. Setting up the IAM/RLS proc from the example provided in the IAM Installation Control Library (ICL), and placing it in a system procedure library.
6. Start the IAM/RLS procedure. This can even replace the IAMSTART procedure in existence today, because starting up IAM/RLS will automatically start up the IAM VSAM Interface if it is not already active.

Once IAM/RLS is active, data sets will be automatically selected for processing by IAM/RLS, based on installation criteria. Most application programs and jobs will not require any changes to code or JCL to make use of IAM/RLS. Data sets will have to be defined as IAM data sets. The circumstances where changes are required include:

1. For batch applications that update a large number of records on recoverable files, it is necessary to utilize the IAM batch syncpoint process. This can be done through use of the IAM Override AUTOSYNCPOINT or by revising the application program. The maximum number of updated / inserted / deleted records to IAM/RLS recoverable files will be limited to the value specified as MAXLOCKS. This limitation is to prevent potential storage shortages and failures in the IAM/RLS address space. This will also aid in preventing a large number of records from being locked out for update by other jobs and CICS transactions.
2. Install and activate the IAM CICS exits that are needed for communicating transaction events to IAM/RLS.

The IAM/RLS address space is designed and intended to provide its services in the background, with little if any operator intervention. However, should the need arise; there are a variety of commands that can be given to the IAM/RLS address space through the Z/OS MODIFY (F) command. These include display as well as action commands. IAM/RLS activity can also be viewed from a TSO/ISPF session, using the provided IAMBMON program. This program can be invoked either as a TSO command from any ISPF panel, or from the IAM ISPF utility menu.
The IAM Record Level Sharing within a single system is achieved by IAM internally passing all of the I/O requests for the files being shared to the IAM/RLS address space. This is done automatically for files that meet the installation’s selection criteria, which can be determined from a combination of share options and dataset names as set by the IAM Global Option of RLS. If the IAM/RLS address space has not been activated, and the RLS Global Option has not been changed, then processing for those files will be performed as it is currently within the job’s address space. The use of IAM/RLS can also be controlled by use of the IAM ACCESS overrides of RLS and NORLS.

IAM utilizes z/OS cross memory services to provide the record level sharing services. All of the OPEN, CLOSE, and I/O requests for shared files will be passed to the IAM/RLS address space for processing, and the appropriate status, key and record will be passed back to the user’s address space when the request has completed within the IAM/RLS address space. Record level locking is achieved within the IAM/RLS address space, with a lock manager that will handle the locking requests, and provides for deadlock analysis and detection within the resources managed within the IAM/RLS address space.

IAM also utilizes the z/OS dynamic resource manager facilities so that IAM will get control whenever a job step or an address space that is using IAM files through IAM/RLS terminates. This function will make sure that the IAM file(s) are properly closed within the IAM/RLS address space will release any record locks for jobs that are normally ending, and will retain record locks of recoverable files for jobs that are abending.

Multiple IAM/RLS address spaces can be executed on any particular LPAR or Z/OS system image. Each IAM/RLS address space must have its own unique 4 character identifier, referred to as the RLSID, must have a separate PROC with a different name than other IAM/RLS address spaces, and must have its own journal datasets and its own set of RLS parameters. The RLSID can be specified via the RLS parameters, or with from the IAM Global Options Table, which can be copied into another load library and having a different RLSID value. Use of the additional IAM/RLS address spaces can be selected by IAM CREATE or ACCESS Override specifications of RLSID values, through the specification of an RLSID within the eligible dataset name table, or by using a different IAM Global Options Table from a separate load library.

The restrictions with using multiple IAM/RLS address spaces are primarily that a job or CICS region can establish a connection with only one IAM/RLS address space, and an IAM dataset can only be opened under one IAM/RLS address space at any point in time.

In addition to handling the I/O and locking services, the IAM/RLS address space can also journal before and / or after images of updated records. The journaling can be performed either to standard DASD sequential data sets, or to the z/OS System Logger. The IAM journaling services are primarily provided to allow for the back out of updates performed by failing batch job(s), or to perform a forward recovery of updates if a file has encountered media damage. CICS will handle its own transaction back out and other recovery as it does today, using its own logging mechanisms that is independent of the journaling provided by IAM.
IAM/RLS journaling facilities can be used for IAM datasets that are not being processed by IAM/RLS. This will provide improved usability for customers using journaling by eliminating a one to one correspondence between a journal dataset and the actual IAM dataset. The journal datasets can be managed at the system level, and only one execution of the IAM journal backout recovery program, IAMJREST, is needed even if multiple IAM data sets are processed by a job. This capability is activated by specification of the IAM Global Option ENABLE=RLSJRN.

**Security**

The IAM/RLS address space will have to be given the appropriate security authority to update those files that are going to be processed by IAM record level sharing. This will include CREATE authority for files that can be extended to additional volumes. IAM does issue the RACROUTE macro within the individual job’s address space to validate that the requesting user does have authority to read or update, as appropriate, the IAM data set being opened prior to requesting that IAM/RLS open the data set. If the RACROUTE indicates that the user / job do not have authority to access the data set, then the OPEN request is failed. The failing job will receive an IAMW18 error message. Additional security will be performed by z/OS services when IAM attempts to take an extent.

**Reliability**

IAM/RLS utilizes the various z/OS error handling and recovery facilities to recover from errors and abend conditions that may occur. The two goals in providing error recovery routines are to provide continuous availability of the services being provided by the IAM/RLS address space, and secondly to automatically collect enough information about any failures such that problem determination and correction can be performed from the single failure. The job that had submitted an I/O request to IAM/RLS that causes a failure will in most circumstances, receive a failure code. It will be up to the program to decide whether to continue processing.

The error data collection will include various messages to the job log, system log, and when applicable the RLS log indicating the abend code, general registers, access registers if applicable, and the failing module. The error routines do attempt to not duplicate diagnostic information that is produced by z/OS, but rather provide additional diagnostic information to be combined with the information provided by z/OS. The error information contained in these messages may be sufficient for problem determination and resolution, particularly if a problem had been previously reported to Innovation. If the error occurred within the IAM/RLS address space, an error trace table will be kept in storage for reference, particularly in situations where multiple errors have occurred. Most error situations will also result in a request for a dump to be taken to a system dump data set, which may include both the IAM/RLS address space and possibly the address space of the job that submitted the failing request. The IAM index space associated with the IAM/RLS address space may also be dumped.

**Serviceability**

To aid in serviceability a mechanism is being provided for system support personnel to apply critical fixes to the IAM modules within the IAM/RLS address space, without the need to shut down the address space. This facility will also be able to back out fixes should they cause other problems.

**Record Locking**

For the single system record level sharing, IAM utilizes its own record level lock manager. The IAM lock manager utilizes a hashing algorithm to provide for fast lock acquisition and release. The trigger to release a record lock will depend on the environment in which the lock was requested. The IAM lock manager will check for potential deadlocks within the scope of the IAM data sets that it is managing. If a deadlock condition would occur the request will be failed with a logical error.
For CICS transactions, records are locked by transaction identification. IAM determines from CICS whether or not the IAM files accessed are recoverable, and if they are it will hold the record lock(s) until either a SYNCPOINT is executed, or the transaction ends. If an IAM file is detected as being not recoverable under CICS, then the record lock is only held from the time of the GET for UPDATE until the record is actually updated or erased, or for records being added, only for the duration of the actual add processing. IAM/RLS also provides support for the CICS VSAM RLS parameters UNCOMMITTED, CONSISTENT, REPEATABLE, and NOSUSPEND. IAM/PLEX provides equivalent functionality for these parameters.

For other than CICS processing, that is batch jobs or TSO users, IAM will generally only hold the record lock(s) from the time of the GET for UPDATE until the record is actually updated or erased, or for records being added, only for the duration of the actual add processing. The exception to this is if IAM is journaling before images of records, implying that a back out could be performed if there is a failure, then the record lock(s) will be held until the program either calls the IAM batch syncpoint, or the job step terminates.

If a job step abends while processing a recoverable file, then any record locks obtained for that job step will be held until a recovery takes place. If there were any other jobs or CICS transactions waiting for the record locks that are being retained until recovery, those requests will be failed when the abend has been detected. The recovery can be performed by IAM’s Dynamic Job Back Out function, by IAMJREST, or by whatever other procedures or recovery software that may be available. Information about any retained locks can be found by using the DISPLAY, RETAINEDLOCKS modify command to IAM/RLS. Such retained locks can also be released by the RELEASELOCKS command.

IAM/RLS provides a Dynamic Job Back Out function for IAM files opened through IAM/RLS. Whenever a job step abends, all updates done by that job step will be automatically removed. If the failing job step has taken IAM batch syncpoints, then the back out will be performed to the most recent syncpoint taken by the job step prior to abending. Control of Dynamic Job Back Out will be provided as an IAM/RLS parameter, and as an IAM Override.
IAM's Real Time Tuning

With the introduction of the IAM VSAM Interface over twenty years ago, IAM introduced a unique buffer management capability called Real Time Tuning. Real Time Tuning is a concept that IAM uses to dynamically manage the buffering for IAM files in response to the processing requirements. This capability allows IAM to adjust the number of buffers being used for any particular file, along with dynamically selecting appropriate buffer management and I/O techniques. This innovative and unique buffering technique has proven itself to provide extraordinary improvements in elapsed time to process batch jobs, as well as reductions in online transaction response times. IAM Real Time Tuning works within a range called MINBUFNO and MAXBUFNO values for the number of buffers, which can optionally be specified on a file-by-file basis by the user.

IAM buffering during a file load does not utilize the Real Time Tuning concept, because the file load I/O process is a sequential output process. The buffering technique used by IAM for a file load is described later in this section.

Benefits of Real Time Tuning

The benefit of Real Time Tuning is the ability to easily achieve a level of performance that is difficult or impossible to reach with VSAM, along with a significant reduction in the manual effort and data gathering required for tuning your indexed files to meet the processing requirements. IAM's Real Time Tuning eliminates the need for deciding which buffering technique to use. With VSAM, you have to choose between NSR and LSR buffering or leave it to DFSMS System Managed Buffering or other packages to make the decision between NSR and LSR. IAM has eliminated the need for index buffers, because IAM automatically keeps the index in virtual storage. For most data sets and applications, the default buffering provides for a very high level of performance. The IAMINFO run time report typically provides more than sufficient information to make any manual buffering adjustments that may be desired. This report, which is produced whenever an IAM data set is closed provided that there is an IAMINFO DD card allocated to the job step, will even indicate with the IAM368 message if more buffers would have improved performance. The IAMINFO report can also be obtained from the IAM SMF records, if they are being collected, with program IAMSMF. In fact, that program can be used to print just the IAMINFO reports for those data sets for which IAM indicated that more buffers would have improved performance. This can be done by using the IAMINFO command with the ATTRIBUTE=MOREBUFFER keyword.

VSAM System Managed Buffering decides for you whether to use NSR or LSR buffering, and will use more buffers than the normal VSAM default. Even with System Managed Buffering, you may still have to indicate through JCL what buffering technique will work best with the way the particular application uses the files. While jobs using VSAM files with System Managed Buffering may perform better than without, System Managed Buffering does not offer the dynamic buffer management that IAM Real Time Tuning offers. In many circumstances, IAM's Real Time Tuning still does noticeably perform better than VSAM with System Managed Buffering.
Real Time Tuning for IAM files includes the following capabilities:

1. As shipped, IAM can have up to 24,576 buffers per file. This capability can be easily increased if necessary.

2. The capability to handle concurrent I/O requests, both at the logical level and at the physical I/O level. This improves IAM responsiveness for online systems, for multi-volume data sets, and for volumes with Parallel Access Volume (PAV) capabilities. IAM’s channel programs are set up to utilize PAV capabilities when they are available for improved I/O responsiveness, which eliminates the need to spread datasets across multiple volumes for performance.

3. Incorporate elapsed time into buffer adjustment algorithms. This will prevent the slow increase in buffers to lightly used files in an online system that is opened for several hours.

4. Provide sequential I/O read ahead, with I/O and processing overlap capabilities for synchronous mode (batch jobs) I/O processing. Up to a full cylinder's worth of data can be read in one physical I/O.

5. For mass sequential updates, IAM will write out multiple updated blocks per physical I/O. This includes the ability to write out multiple blocks per I/O even if they are not contiguous.

6. IAM buffers can reside in 64-bit addressable virtual storage or in 31-bit addressable storage based on Global Options and overrides. Should a buffer end up in 24-bit addressable virtual storage, IAM will release that buffer, to help avoid a potentially critical virtual storage shortage, unless that is the only buffer that IAM has available.

7. For Share Option 1 or 2 files, IAM also has incorporated empty block detection, which will prevent subsequent physical I/O for such empty blocks. This eliminates a problem encountered with rereading empty blocks when there had been large quantities of records deleted from the file.

8. The IAM I/O driver will also take advantage of the IBM Record Level Caching, for random I/O's and indicate sequential processing on the devices that provide those capabilities.

9. IAM channel programs will automatically utilize z/HPF High Performance Ficon channels using the Transfer Command on the volumes that support the architecture. IAM continues to provide optimal use of the latest hardware in the market, by using block sizes that fully utilize the device’s capability, and by using the latest features and capabilities when performing I/O.

10. A TURBO mode, which defaults to being enabled, is provided to allow for faster acquisition of buffers during periods of heavy I/O activity.

For random processing, IAM dynamically acquires additional buffers up to the MAXBUFNO value, so that the most frequently referenced blocks can be retained in memory. IAM continually monitors what the added benefit would be for additional buffers at any point in time by tracking contents and frequency of use. By monitoring requests in this way IAM can recognize patterns. After certain predetermined numbers of physical I/O’s, IAM will evaluate the buffering. When IAM finds that a sufficient number of requests could have been satisfied without I/O to the disk within the past few minutes if more buffers had been available, an additional buffer is acquired. If the file is in Turbo Acceleration Mode up to four buffers may be acquired at one time. This continues until the maximum number of buffers allowed is reached.

IAM's buffer management techniques also provide for the release of buffers when it determines they are not providing any benefit in terms of reducing I/O within the past several minutes.
Buffer management for sequential processing differs from random in that IAM attempts to determine how many blocks it should be reading ahead in anticipation, to efficiently service the user. The goal is to be able to read as many useful blocks as possible with each real disk I/O. To accomplish this IAM monitors the number of blocks that are read consecutively, increasing or decreasing the number of consecutively chained buffers as appropriate. A maximum of up to one cylinder worth of blocks will be read per I/O. More buffers are acquired as necessary, at the rate of at most one per physical I/O, up to the MAXBUFFNO value. Once processing has been completed for each block read sequentially, rather than aging the block out of the buffer pool, it is immediately treated as being the buffer with the least recent reference, because it is not expected that the block will be needed in the near future. For synchronous sequential I/O, which is typically done by batch jobs, when reading multiple blocks per physical I/O, IAM will make the data from each block available to the application program as soon as it is in storage. On mass sequential updates, when multiple consecutive data blocks have been updated, IAM will write out multiple blocks per physical I/O. This process is done asynchronously whenever possible, to allow for I/O and processing overlap.

The advantage of reading or writing multiple blocks per I/O is improved performance due to a reduction in resource utilization. The primary savings are at the device level. This type of I/O will eliminate DASD seek and search time, as well as rotational positioning time because the device is already positioned to process the next block. This device time can frequently be well over 50% of the total device time needed when performing a single block I/O. Also eliminated is the CPU overhead involved with issuing an EXCP, scheduling the I/O, plus the CPU time involved with the I/O completion. Also, for batch jobs, IAM will attempt whenever possible to allow the sequential I/O to be active while allowing continued processing of the data that is already residing in IAM's data buffers.

Multiple random and / or sequential I/O operations can occur concurrently against any file. The use of the buffers will be balanced between the different requests based on the buffering needs of each request. The request with the highest buffer requirements will tend to utilize the most buffers. IAM does not lockout a request from a data block if another request is also using the block, such as VSAM does with the CI level lockout. A request may be temporarily delayed access to a block of data only if there is an actual physical I/O occurring, or during the actual processing time to move a record into or out of the buffer. This delay is managed internally by IAM, and does not result in deadly lockouts. The CI lockout problem caused by VSAM resulted in many installations not being able to use VSAM LSR buffering for some files under CICS, or forced files to small CI sizes to minimize the occurrence of the lockout problem. IAM eliminates these VSAM problems. IAM does perform the necessary record level lockout for concurrent updates to the same record to maintain data integrity.

For IAM data sets defined with Share Option 1 or 2 that are opened for update, IAM does distinguish the differences between synchronous and asynchronous processing. Synchronous processing is indicated by the RPL OPTCD=SYN, and is typically used by batch jobs. Asynchronous processing is indicated by the RPL OPTCD=ASY or OPTCD=WAITX, which is typically used by online systems, such as CICS. To improve batch performance IAM defers the rewrite to disk of a randomly updated buffer until the block residing in the buffer is forced out of the buffer pool. This technique reduces I/O whenever multiple rewrites are requested against data blocks in the buffer pool. Under Online systems (e.g. CICS), to insure integrity after a random update, buffers are always immediately rewritten. For sequential processing, the updates are deferred regardless of processing mode until IAM has several blocks to chain write, until the caller issues an ENDREQ request, or until the buffer is needed for another data block.
With sequential synchronous I/O, IAM is very aggressive with increasing the number of blocks that are being read per I/O, as well as being very quick to increase the number of buffers up to MAXBUNO. This is done to improve elapsed time for batch jobs. When asynchronous type of I/O is being done, IAM is less aggressive with sequential chaining. This is done to prevent tying up too many buffers at one time for online systems. This will prevent issuing an I/O request, which will monopolize a device for a relatively long period by reading many blocks per I/O, which could impact the responsiveness of the online system for other transactions.

IAM's Real Time Tuning works within an optionally specified range for determining the number of buffers called MINBUFNO and MAXBUFNO. MINBUFNO defaults to 1, unless explicitly overridden on an IAM Override statement. The value for MAXBUFNO can come from a number of sources, as indicated below in precedence order:

1. From the IAM ACCESS override of BUFSP. If provided, this value always has precedence over any other specification.
2. From the IAM ACCESS override value specified for MAXBUFNO.
3. From the IAM CREATE override value specified for MAXBUFNO when the file was loaded or defined.
4. The buffer space value specified on the IDCAMS DEFINE of the data set, providing that it is greater than the IAM Global Options BUFSP value.
5. The higher of BUFND, BUFSP, or STRNO specified in either the JCL AMP parameter or explicitly in the ACB, provided that it is larger than the default MAXBUFNO determined from the IAM Global Options Table. (Note that JCL specification of one of these values takes precedence over the value specified in the ACB control block itself.)
6. A default value from the IAM Global Options Table, which will be the higher of either the MAXBUFNO Global Option, or the BUFSP Global Option (CICS will use CICSBUFSP) value divided by the file's block size.

For a typical 1/4 track blocked IAM data set on a 3390 device, using the default Global Option values, of 65,536K for batch, and 1,024K for CICS IAM will of set the MAXBUFNO value to 4904 for batch, and 76 for CICS.

When an IAM data set is opened, IAM will determine the values that it will use for MINBUFNO and MAXBUFNO as described above. If MINBUFNO value exceeds MAXBUFNO, it will be reduced to the MAXBUFNO value being used. IAM will then decide on a starting value. For batch jobs, IAM will initially acquire the higher of number of buffers required for 16 tracks or MINBUFNO if it was specified as an override but will not exceed MAXBUFNO. For CICS IAM will initially acquire the smaller of the number of buffers to read the index or needed for 15 tracks but will not exceed MAXBUFNO. Under CICS, the number of buffers will be reduced at completion of open processing to the number of strings (STRNO) or if overridden the MINBUFNO value.

As a general rule, there should be very little necessity to specify a MINBUFNO value. With the TURBO mode buffering enabled, buffer acquisition is very fast for up to 80% of the MAXBUFNO value. MAXBUFNO can be changed as desired, and generally should be increased when IAMINFO indicates that more buffers will be helpful, and you need to improve the performance for the data set.
The IAMINFO report, which is produced when an IAM file is closed, will provide a variety of statistics relating to IAM Real Time Tuning and its effectiveness. These fields include the following:

**DISK BLOCKS READ** - Indicates the number of physical I/Os (EXCPs) that were done to read data into the IAM buffer pool.

**DISK BLOCKS WRITTEN** - Indicates the number of physical I/Os (EXCPs) that were done to write data out to the file.

The TOTAL EXCP's issued by IAM for this data set can be calculated by adding the DISK BLOCKS READ plus the DISK BLOCKS WRITTEN. That number will be equal to the EXCP value reported by SMF for this data set, except for a file load with a temporary work data set. For a file load with a temporary work data set, the IAMINFO values are inflated by the I/O's done to the temporary data set.

**DYNAMIC BUFFER RETRIEVALS** - Displays the number of I/O requests satisfied from the IAM buffers without performing a physical I/O. This statistic essentially identifies the savings of I/O due to IAM's Real Time Tuning.

**MAXIMUM BUFFERS USED** - Indicates the maximum number of buffers that were used.

**MINIMUM BUFFERS USED** - Indicates the smallest number of buffers that were used.

**MAXIMUM BUFFERS AVAILABLE** - Indicates the MAXBUFNO value in effect for this file.

**SEQ CHAINED BLOCKS READ** - Indicates the number of data blocks that were read in as part of a sequential multiple blocks per I/O. These blocks were read as part of IAM's Real Time Tuning anticipation that the application program will use the blocks. This value plus the DISK BLOCKS READ value indicates the total number of blocks physically read into storage for this data set.

**SEQ CHAINED BLOCKS WRITTEN** - Indicates the number of data blocks that were written as part of a sequential multiple block output I/O. This indicates the savings of physical I/O (EXCPs) due to IAM's Real Time Tuning accumulation of updated data blocks. This value plus the DISK BLOCKS WRITTEN value equals the total number of blocks written out to the data set.

**z/HPF I/O REQUESTS** - Indicates the number of EXCP's issued using the High Performance Ficon architecture I/O. This value is provided so users can verify IAM's utilization of the High Performance Ficon.

**ECKD I/O REQUESTS** - Indicates the number of EXCP's issued using standard ECKD or CKD I/O architecture. This value may be non-zero even when z/HPF is being utilized for I/O issued for writing the DASD EOF, new extent processing, recovery from an I/O error, or when z/HPF has been temporarily turned off for a volume in use.

While all of the above information is useful in evaluating the benefits of IAM's Real Time Tuning, there may still be a question as to whether or not more buffers would have been useful. The IAMINFO report will come out and tell you via an IAM attention message, message id IAM368 that is displayed in the run time INFO report. A flag is also stored in the IAM SMF record.
The IAM368 message is just an indication that the I/O performance for the specified file could potentially be improved by providing a larger MAXBUFNO value. It may very well be that the performance level of the job processing this data set is quite acceptable, and if so there is no need to increase buffering. However, if you are looking for ways to improve the performance, then this message is an indicator of one of the possibilities for improving performance.

If the IAM368 message does not appear, then providing more buffers is very unlikely to change the performance. However, for some applications that are very randomly processing the file, it may be beneficial to provide larger values for both MINBUFNO and MAXBUFNO, in case the reference pattern encompasses more blocks than what IAM’s Real Time Tuning has been tracking so it hasn’t been increasing the number of buffers.

For more information on tuning and getting the most out of IAM, refer to the Tuning Section 10.50 in the IAM User's Guide portion of the manual.

File Load Buffering

As indicated at the top of the Real Time Tuning section, a file load does not make use of the Real Time Tuning concept. This is because a file load is exclusively a sequential output process. A fixed number of buffers are acquired at open time, based on the value for the CRBUFOPT Global Option or IAM Override. The required channel programs and I/O control blocks are built and initialized during open based on the quantity of buffers acquired. By default, for files that are allocated on cylinder boundaries, IAM will use enough buffer space to hold two cylinders, and write out a full cylinder per physical I/O. The default for track allocated files is buffer space for two tracks, and writing out one track per I/O. This buffering concept provides for I/O overlap, even for programs using synchronous processing. The EXCP counts will reflect the actual EXCP (physical I/O) operations done not the block count.

I/O processing during file load is different for DFSMS Extended format datasets. While the number of buffers will be the same as for EXCP processing described above, IAM is using BSAM access method to write out the data to the extended format datasets. IAM will allow BSAM to write out multiple blocks per I/O. However, with BSAM output, it is not possible to control how many blocks will be written out per physical I/O. Also, because BSAM is being used, the EXCP count for file loads will be the actual number of blocks written out to the dataset, not the number of physical I/O requests. So, even though multiple blocks are being written per I/O, that saving is not reflected in the EXCP counts.

z/HPF

High Performance Ficon (z/HPF) is an IBM architecture for physical I/O that enables more efficient utilization of I/O channels resulting in improved performance and greater capacity. This function has recently become available for IAM utilization with hardware and software enhancements that enable utilization by software that uses EXCP level I/O. Based on the results at Innovation, we see a 20% to 25% reduction in channel connect time when IAM uses z/HPF, and a small improvement in elapsed time.

The z/HPF support requires IAM Version 9.2 along with z/OS 2.1, or z/OS 1.12 and above with enabling PTF. On the hardware side, it requires DASD subsystem support of the functions provided by z/HPF for EXCP. IAM validates eligible devices through the use of the IOSZHPF service. Users should contact their DASD subsystem vendor to determine what is necessary for EXCP I/O support.

64-Bit Virtual Storage Buffers

IAM Version 9.2 can optionally utilize buffers in 64-bit virtual addressable storage. This is primarily of benefit for CICS, IAM/RLS, and IAM/Plex address spaces where there are 31-bit addressable storage constraints. This may cause an increase in CPU time due to page fix processing for 64-bit virtual storage.
IAM Data Compression

Advantages

Data compression reduces DASD disk space usage and reduces both the amount of data transferred over the I/O subsystem which reduces both physical I/O time and the amount of physical I/O. IAM can compress the data in KSDS, ESDS, RRDS and AIX files. IAM compression is completely transparent to the programs that create and use those files. Customers can choose to use either IAM software compression, or they can request that IAM use the z/Architecture hardware compression instruction. With hardware compression, users can choose to either have IAM dynamically build the compression dictionary during the file load, or they can build a customized dictionary prior to actually loading the dataset. The primary advantage to building a customized dictionary is that it will generally provide the best amount of space savings from data compression.

For software compression, IAM uses a proprietary algorithm that is optimized for minimal CPU processing requirements that also provides good space savings for most files. IAM's software data compression does not rely on a compression dictionary which reduces the exposure to potential data loss. While it is possible for alternative data compression algorithms to achieve a greater amount of record size reduction, the CPU time to achieve such results can be excessive. In fact, for many files, IAM's software compression CPU time is still less than VSAM's CPU time without data compression! Also, because of VSAM's use of generic compression tables, IAM still, for many files, is able to achieve similar or better space savings than VSAM hardware compression.

IAM's CPU time with hardware compression will be less than VSAM CPU time with hardware compression, and with customized compression dictionaries, IAM may achieve greater compression. Unlike the IAM software compression, the IBM hardware compression requires the use of a compression / decompression dictionary. IAM will generate a dictionary using up to eight megabytes of data from the first records initially loaded into the dataset. The intent is to provide a decent level of compression while minimizing the CPU time used to generate the dictionary. Users can build their own compression dictionaries using a more extensive process that will likely yield much greater compression for most datasets. Instructions on creating customized compression dictionaries are in Section 10.25 of the IAM manual.

The default compression technique, of either Hardware or Software, can be set in the IAM Global Options Table by the COMPRESSTYPE option. The default, as IAM is shipped is Software Compression. The general performance with using data compression is that it will increase the CPU time to process a file than without any compression. When using data compression, the consideration is that the hardware compression will use more CPU time on the file load or reorganization than the software compression, however hardware compression will use slightly less CPU time for normal dataset access. For files that are very infrequently loaded or reorganized hardware compression will generally be using less CPU time.

Eligibility for Data Compression

IAM considers KSDS and AIX type of files that are 75 tracks or larger, and all ESDS and RRDS files as candidates for data compression. Setting the IAM DATACOMPRESS Global Option can easily change the file size that determines eligibility for IAM Data Compression. The automatic setting of IAM's Data Compression can always be altered through the use of the IAM Override facility.
To be eligible for IAM Data Compression, a file must be defined with a maximum record size that is at least 10 bytes more than its key length plus the relative key position (RKP). If Data Compression is enabled for a file, IAM will only compress individual records when the data following the record key exceeds ten (10) bytes in length. If compression would make a record larger than the original, IAM leaves the record uncompressed. Subsequent updates to an uncompressed record will keep the record uncompressed.

Data Compression can be used on all your IAM files. If a particular file is found by IAM to be uncompressible, there is no penalty in CPU time to process that file after the load. It is as if compression had never been requested for that file. There may also be a few files that just do not show much of a benefit from data compression. For example, SMP/E CSI files have an average record length that is just a bit larger than their key. When there is not much data to work with, there is little data compression can do to reduce a file’s size. In these cases IAM’s Data Compression may show little saving, beyond the space reduction that comes with simply converting to IAM. If a specific file shows only marginal compression there will likewise be only a marginal increase in IAM’s CPU time to process that file.

IAM offers the capability to backup and reload software compressed data within an IAM file without decompressing or compressing the data. For large files, this is anticipated to allow IAM files to be backed up and reorganized faster than can be done today. Even when the data is compressed by the tape control unit, there is still the overhead of transferring all that data to the controller. With this new feature, both the CPU overhead and that I/O overhead is eliminated. The FDRREORG product from Innovation will automatically use this IAM feature.

The backup and reload of software compressed data is specified for other programs, such as IDCAMS, by the use of the IAM Override facility. The override will have to be specified on both the backup and reload process, because IAM needs to know to not decompress the data on the backup side, and that the data is already compressed on the input side. Simply specifying the keyword BACKUPCOMPRESSED on the ACCESS and CREATE IAM overrides does the job. IAM adds four bytes to each record when performing this function, so any output file created will have to contain either variable (RECFM=VB) or undefined (RECFM=U) type of file. For variable output files, the record length for the output file (LRECL) will need to be at least 8 bytes more than the defined maximum record length for the file. For example, if the maximum record length for the file is 100, then the output LRECL must be at least 108. For undefined type of records, the maximum LRECL is 104, only 4 bytes more than the file maximum record size. Innovation recommends using RECFM=VB type of output to provide the best output device utilization.

Data that is in an IAM data compressed format on tape can be easily converted to an uncompressed format. Either reload the data with BACKUPCOMPRESSED into an IAM file, or use the IAMRECVR DECOMPRESS command to make a sequential copy of the data set with uncompressed data. Note that you will need to know the original key length and key offset (RKP) to perform the DECOMPRESS function. IAM also provides a callable interface to read and perform the decompression from a data compressed sequential data set that can be used by application programs.

For examples of using the BACKUPCOMPRESSED feature and using FDRREORG to reorganize IAM data sets, refer to Section 10.81.
IAM Journaling

IAM provides an optional automatic journaling capability on file updates for IAM files. This capability is provided to assist in improving data availability, particularly for the very large multivolume data sets. The improved availability of the data is accomplished by reducing the frequency of data set backups, and providing a speedier recovery facility for failing batch jobs.

An example of the intended use is as follows. Let’s say there is a multivolume IAM data set that is updated every day, from both online and batch. Because of this, the data set is backed up every day. With IAM journaling AFTER images, the file may now need to be backed up only once a week; however the smaller log data set containing the updates will be backed up daily. If a recovery is needed, the data set is first restored from the last backup, and subsequently all of the updates are reapplied from the IAM journal to the data set using the IAM journal recovery program IAMJREST. The data availability is improved by elimination of the time spent backing up the data set every day, which may take well over a couple of hours.

The other intended use is to provide a backout mechanism of updates performed by batch jobs. To utilize the backout capability, the user must indicate that the BEFORE images of updated records are to be included in the journal. Should a batch job abend, rather than restoring the file and rerunning prior batch updates, all that needs to be done is to backout the updates from the batch job step (or entire job, and other jobs). This is accomplished by using the IAM journal recovery program, IAMJREST.

The IAM journaling feature is activated through the JRNAD override keyword, with the appropriate value of BOTH, BEFORE or AFTER indicated for the types of journal records required. As stated above, the AFTER images are necessary for performing a recovery from a data set that has been restored. BEFORE images are needed when backing out data set updates. The BOTH keyword enables collection of both the BEFORE and AFTER images. When specified on the CREATE override during a file definition or load, IAM will automatically journal all updates made to the data set. When specified during file updates on the ACCESS override card, the specified journal options are only applicable to that job step. Additionally, the user must allocate a sequential data set to contain the journal records, with an adequate amount of space. The name of the data set must be the same as the IAM data set name (i.e. cluster name) appended with ‘.LOG’. If the data set name is over 40 characters long already, and if there is a ‘.’ in position 40, then the data set name will be 43 characters long, ending with the ‘.LOG’ string. If the data set name is over 40 characters long, and position 40 is not a ‘.’, then the log file name will be 44 characters long, ending with the ‘.LOG’ appendix. The user is also responsible for the management of the IAM journal data sets). They must be backed up as necessary, and also emptied out at various points. The log data set can be emptied by deleting and reallocating the data set, or by using an IEBGENER to copy into the log data set from a DD DUMMY.

IAM journaling is not intended as a substitute for CICS journaling and the CICS transaction backout capability. Those capabilities must still be utilized if required.

Full information and examples on using the IAM Journaling capability are provided in Section 10.88 of the User’s Guide.
SMS Support in IAM

Overview
IAM provides support for SMS that is equivalent to the VSAM support, including support for JCL file definition, and temporary data sets. By definition, to be eligible for an SMS managed volume, the file must be assigned a Storage Class. The Storage Class, along with optionally a Data Class and/or Management Class, can be explicitly specified on the DEFINE command, by JCL parameters for JCL defined files, or selected by the ACS routines. IAM files on SMS managed volumes will be cataloged with the class names. As a part of the SMS support, an additional method of triggering an IAM DEFINE is available. Files will be defined as IAM files if the Data Class or Storage Class name contain the $IAM literal.

SMS Automatic Class Selection Routines
For both IDCAMS DEFINE’s and VSAM JCL allocations, the ACS (Automatic Class Selection) routines are called prior to the IAM DEFINE intercept. When IAM intercepts the request, the SMS classes, the SMS Storage Group, and the SMS volumes have already been selected. IAM will then screen the request and determine if the file should use the IAM format. If $IAM, in the Data Class (DATACLAS) or Storage Class (STORCLAS) name, is being used as the criteria for determining IAM format files, then the class name(s) must have $IAM in them at this point in the process. They contain either the explicit names from the DEFINE request, or the name(s) selected by the ACS routines. This allows the installation the possibility of controlling IAM files, and IAM usage through the ACS routines. For JCL allocation, these are the classes and volume(s) that will be used.

For IDCAMS DEFINE requests, the ACS routines will be re-entered when IAM issues the dynamic allocation of the file as a non-VSAM data set. The request will specify the SMS classes as received from the intercepted DEFINE request, and the volume(s) that had initially been selected by SMS. On the second pass through the ACS routines, IAM files can be identified checking the DDNAME field for a value of @#$IAM. The ACS routines can change the SMS classes and the Storage Group, which will change the volume(s) on which the file is placed. At this point, the file must not be switched to a non-SMS volume, because the allocation will fail. However, it can be switched from an unmanaged volume to an SMS managed volume. Changing the DATACLAS at this time will have no effect on the file characteristics, as they were determined by IAM prior to the dynamic allocation. The MGMTCLAS, STORCLAS, and Storage Group can all be effectively changed by the ACS routines on the dynamic allocation request.

SMS Classes
IAM provides full support for IDCAMS defines under SMS. The DATACLAS, STORCLAS, and MGMTCLAS can either be explicitly provided on the DEFINE command, or selected by the ACS routines. The Data Class can provide file characteristics for the file being defined, including record length, key length, key offset, share options, free space, and others, eliminating the need to specify those values explicitly on the DEFINE. As per SMS rules, the options in the Data Class will be used, unless explicitly overridden on the DEFINE command.
**DFSMS Extended Format**

IAM files can optionally be defined as DFSMS extended format sequential datasets. These datasets can fully utilize larger volumes, such as a 3390-9 or even larger volumes of up to 64K cylinders. With z/OS 1.11 or higher these datasets can fully utilize EAV volumes, those exceeding 64K cylinders. These datasets can use up to 123 extents per volume and exceed the prior limit of 255 extents in total when multiple volumes are specified. Such data sets must be DFSMS managed, with a data class that specifies “Extended Required” or “Extended Preferred” for the data set name type, and with Compaction set to NO. These datasets must also have a Storage Class with a “Sustained Data Rate (SDR) either left blank, or set to 0. Setting other values for Sustained Data Rate will cause the dataset to have multiple STRIPES, which IAM does not support. IAM will internally compress the file, with either software or hardware compression.

**Allocation Errors**

If the IAM allocation encounters any errors, the error messages will appear on the JES job log, with the Z/OS allocation messages (SYSMSGS) and also on the IDCAMS SYSPRINT, if it is available. Due to the manner in which IDCAMS prints messages on SYSPRINT, the error messages from IAM will precede the actual DEFINE command. IDCAMS will also print out additional error messages after the DEFINE, performing an analysis on the return codes set by IAM. Whenever possible, IAM uses the VSAM return codes that most clearly indicate the actual problem, although that is not always possible. Always refer to the IAM and related allocation error messages for the most precise problem determination possible.

**JCL Allocations**

VSAM files being defined through JCL can also be easily converted to IAM files. This is done by either putting $IAM in the data set name (DSN), or by using a Storage Class (STORCLAS) or Data Class (DATACLAS) with $IAM in the name. Both permanent and temporary data sets can be defined, with the restriction that temporary data sets cannot be multivolume, same as with VSAM. The use of a Data Class (DATACLAS) is highly recommended for JCL defined files. By using a Data Class, values for Free space, CI Size, and Share Options can be specified, which are not available through JCL parameters. All files defined in JCL will, by default, be capable of handling variable length records, up to the maximum length specified in the DATACLAS or LRECL field.

To allocate IAM files through JCL, IAM must be in the link list. STEPLIB and JOBLIB are ineffective in this case, because it is the initiator that is issuing the allocation, and IAM must have access to various load modules for the define.

If the IAM allocation encounters any errors, the error messages will appear on the JES job log and with the Z/OS allocation messages (SYSMSGS). SMS will also print out additional error messages appearing with the Z/OS allocation messages, performing an analysis on the return codes set by IAM. Whenever possible, IAM uses the VSAM return codes that most clearly indicate the actual problem, although that is not always possible. Always refer to the IAM and related allocation error messages for the most precise problem determination possible.

Any CREATE overrides for JCL allocated files must be in the job step that loads the file, not necessarily the step allocating the file. The define process does not access the IAMOVRID DD for JCL defines. For example, if the file is allocated in an IEFBR14 step, and then subsequently loaded by an IDCAMS REPRO, the IAM create overrides must be in the IDCAMS REPRO step.
**Dynamic Allocation**
The TSO ALLOC command, and the Z/OS DYNALLOC service, has also been enhanced to provide for allocation of new and temporary VSAM files. These requests will also be screened by IAM, and can be converted to an IAM file in the same manner as a JCL allocation can. IAM treats the request just like a JCL request. The new ALLOC keywords are the same as the new JCL keywords, and dynamic allocation has the equivalent text units.

**Multivolume**
IAM files can be spread across multiple SMS managed volumes, both with and without Guaranteed Space. Note that IBM restricts temporary VSAM files to a single volume, this also applies to IAM. When an IAM file is defined with a Storage Class that specifies Guaranteed Space, the primary allocation quantity is allocated on each volume at DEFINE time, as per the SMS non-VSAM rules. When Guaranteed Space is NOT specified, only the first volume is selected at DEFINE, and the subsequent volumes are cataloged as an '*'. During file load or reorganization, if additional volumes are needed, they will be selected by SMS. SMS has a restriction that within a job step, if a file defined without guaranteed space uses additional volumes, only one DD statement can be used, because any other DD’s are not updated to indicate the additional volumes.
Automatic DASD Space Functions

When IAM files are defined with a secondary space value, IAM will automatically release over allocated space after the first load of an IAM file.

If IAM were to release all of the unused DASD space at the end of the file load, a data set would very quickly end up taking extents as the file expands. Rather than eliminate all of the unused space, an automatic space reserve feature was developed. With the space reserve feature, some of the over allocated space will be retained rather than released at the end of the first load to allow for file growth without having to immediately go into secondary extents. The space reserved will be contained within the allocated extents after the file has been loaded. No additional extents will be obtained for the space reserve.

The desired amount of space to reserve will be calculated by using either the value specified for Overflow records on a CREATE IAM Override card or by using the CA% free space value specified on the Define of the file. Basically, the CA% free space value is cut in half, and then that percentage of the total DASD space required for the data set as loaded will be set as the desired reserve quantity.

When the program loading the file issues the close, IAM looks at how much DASD space is allocated to the data set, but is not yet used. If the allocated but unused space is less than the desired reserve quantity, then no space will be released. However, if the allocated but unused space is greater than the desired reserve quantity, then space will be released down to the desired reserve quantity. In all cases, the end of file is set at the end of the used area, and is indicated so in the VTOC LSTAR field. This will allow a DASD space management utility, such as Innovation’s FDR/CPK product, to release the unused space that was reserved if it has not yet been used.

For example, consider a file where CA% Free space is defined as 20%, and the file ends up using 300 tracks of space. A DASD reserve value of 30 tracks is calculated by taking half of the CA% free space, or 10% of the 300 tracks. Any allocated space over 330 tracks will be released. If the data set is allocated to less than 330 tracks, then no space will be released. If the file was allocated initially with 600 tracks, then 270 tracks of space will be released, leaving a total of 330 tracks allocated to the IAM data set.

If CA% free space is 0 and there is no Overflow override, then no space reserve will be done, and all excess allocated space will be released. In the example cited above, the data set would end up with only 300 tracks allocated. If the desired space reserve quantity is less than one cylinder, then no space reserve will be done, because most files are cylinder allocated, and the space release is done at a cylinder boundary.

RELEASE is an especially important IAM feature. IAM files generally take 30 to 70% less disk space than VSAM. If the original VSAM IDCAMS DEFINE space allocation values were left unchanged without auto- RELEASE, a lot of disk space would continue to go to waste.

RELEASE is IAM’s default and as most other IAM defaults it can be changed in the IAM Global Options table using the program IAMZAPOP, see Section 91 of the IAM manual.
IAM will dynamically adjust the secondary space quantity for IAM files that are in standard non-extended format or in Large Sequential format. This function is provided because such datasets are limited to 16 extents per volume. The algorithm will take effect once a file has used five extents on a volume. At that point in time, IAM will increase the secondary space allocation by a multiplication factor specified in the Global Options Table, or from IAM Overrides. The default secondary space multiplication factor as IAM is shipped is ten for file loads, and five for file updates. The Secondary Space Adjustment feature is subject to the following rules:

The secondary space quantity will not be increased to a value that exceeds the original primary space quantity.

The secondary space quantity will not be increased to a value that exceeds the size of the largest contiguous available extent on the volume. Note this is only effective once the file has obtained five extents.

If the original secondary space quantity is higher than the original primary space quantity, the secondary space quantity will not be adjusted.

For single volume files, the original secondary may be decreased from the original value to the largest extent available on the volume, just to try to keep it running as long as possible.

Secondary space adjustment is not used for DFSMS Extended Format datasets because they are able to obtain up to 123 extents per volume instead of 16.

Secondary space adjustment of DFSMS managed datasets may yield unanticipated results, particularly for multi-volume datasets. It appears that during the process of extending to an additional candidate volume that DFSMS wills top picking up revised secondary quantities from the JFCB. Subsequent job steps may result in this feature working again, until another volume is added.

For example, if a file is Defined with a primary quantity of 500 cylinders, and a secondary of 10 cylinders, after five extents have been acquired, the secondary during a file load will be adjusted up to 100 cylinders. During a file load, the maximum space that will be used for this file is:

- 1 extent of 500 cylinders
- 4 extents of 10 cylinders each
- 11 extents of 100 cylinders

This will bring the total maximum space up to 1,640 cylinders, which is slightly less than the VSAM maximum of 1,720 cylinders. Although IAM allocated less DASD space than VSAM, the amount of user data kept in the IAM file will generally be larger due to IAM’s space savings capabilities. This is due to IAM’s efficient use of DASD devices and Data Compression feature. So, by providing the Dynamic Secondary Space Adjustment feature, IAM files have the potential to grow in size as large as VSAM files will, although it will be done in fewer extents.
For this same allocation, the secondary will be adjusted up to 50 cylinders during a file update run. However, if the primary space is 20 cylinders, and the secondary is 10 cylinders, then the maximum value that will be used for the secondary is limited to 20 cylinders. The maximum secondary space quantity that will be requested for any particular file is included on the IAMPRINT LISTCAT output.

The secondary extent multiplication factor can be changed by IAM Overrides on a file-by-file basis, using the MAXSECONDARY keyword. Regardless of the value set, the basic rules for modifying the secondary space quantity remain as explained above. The factor can be specified on the IAM CREATE override during the file definition or file load, and on file updates with an IAM ACCESS override. The values permitted are from zero to ten. Values of zero or one will prevent IAM from increasing secondary allocation. The secondary allocation value may be reduced for single volume files when there is insufficient space for the secondary. When the secondary factor is specified on a CREATE override during file definition, the value is stored with the file. That value will be used for subsequent file loading and file updates, unless overridden at run time.

With the IAM Dynamic Secondary Space Adjustment feature, there is an additional option for multivolume files that are defined without guaranteed space. Again, this function does not apply to DFSMS Extended Format IAM datasets. When it appears to IAM that the EOV request will result in the next volume being allocated, then IAM will request the primary space value, rather than the normal secondary quantity that is used for non-VSAM files. This feature can be controlled by either IAM Overrides, or by the IAM Global Options table. This feature is provided to offer an alternative so that IAM space allocations will be similar to VSAM. The IAM Override keyword is:

- MULTIVOLUME=PRIMARY
- MULTIVOLUME=SECONDARY

These IAM Override keywords can be used on the CREATE Override statement. When specified on the CREATE keyword during file definition, the value specified is saved with the file control information and will remain the applicable option, unless overridden by a particular job step.

The rules for IAM files defined on DFSMS managed volumes with the Guaranteed Space attribute are different. This is because DFSMS will automatically allocate the primary space quantity on each volume when the file is defined. The secondary allocation quantity will be modified as described above for single volume files.

Another special multivolume circumstance is when a file is defined with no secondary space specified. For most circumstances, IAM will set the secondary allocation value to the primary value. For data sets on DFSMS managed volumes with Guaranteed Space, the secondary is left as zero, with the primary being allocated on each volume when the file is defined. For systems where DFSMS is active, but the data sets are on non-SMS managed volumes, IAM will leave the secondary as zero, and allocate the primary space on each volume when the file is defined. This is to mimic the DFSMS Guaranteed Space, and will prevent any secondary extents from being taken.
IAM files can contain spanned records. A spanned record is one that is larger than the physical block size of the IAM data set, and will require multiple physical blocks to store it. With IAM, the record size limitation is the amount of data that IAM can fit within 256 data blocks. With a maximum block size of 32760, an IAM file can now have a maximum record length of approximately 8 megabytes. Spanned record support will allow IAM Alternate Index customers to have large numbers of records in a base cluster to have the same alternate key. The spanned record support also eliminates the DASD space waste that occurs on files where the record size requires a physical block size that exceeds the half-track block size that resulted in IAM using a block size of 32760.

Please be aware that many utilities used to process IAM (and VSAM) files do not support record sizes of approximately 32760 bytes or larger, including IDCAMS and the various SORT utilities. The IAM file recovery utility program, IAMRECVR, does provide support for these large records.

One of the features of IAM that helps deliver outstanding performance is to have the index to open files reside in virtual storage. The virtual storage used for the index can come out of any one of the following areas:

1. 64-bit addressable virtual storage, in users address space
2. z/OS Data Space, maximum size = 2 Gigabytes
3. 31-bit addressable in users address space, maximum size: available storage < 2 Gigabytes

The default is to use 64-bit addressable storage if it is available. To make this storage available users will need to specify a MEMLIMIT=nG on their EXEC card unless their installation has set up the exit to automatically allow use of 64-bit virtual storage. The advantage of 64-bit virtual is that it eliminates the 2 gigabyte size limit that a data space has, and keeps the index out of storage areas typically used by application programs. If 64-bit addressable storage is not available, then IAM will attempt to acquire a z/OS data space to hold the index. If IAM is unable to acquire data space storage, then virtual storage from within the users region will be utilized. The default can be changed either by modifying the INDEXSPACE Global Option or by IAM overrides for the particular job steps.

IAM has support for 8-byte RBA values in the IAM ESDS type of files. This is referred to by IBM as Extended Addressability. VSAM ESDS datasets defined with this DFSMS attribute can exceed 4 gigabytes by using an 8-byte RBA, instead of the normal 4-byte RBA. An IAM 8-byte RBA file is created by specifying the XESDS keyword on the IAM CREATE override, through specification in the DFSMS Data Class, or by setting the IAM Global Option ENABLE=XESDS. Such IAM ESDS files do not have to reside on DFSMS managed volumes, as do VSAM ESDS files. Your application programs may not be able to take advantage of the 8-byte RBA values without coding changes. For such situations, you may find that the IAM PSEUDORBA feature, which provides for over 4 gigabyte ESDS files with a non-standard 4-byte RBA as an easy solution.

PSEUDORBA type ESDS files are created through the PSEUDORBA IAM CREATE Override. This enables many applications to exceed the 4-gigabyte size limitation without changing to an 8-byte RBA value. The limitation is that if an application is dependent on the RBA value being identical to the VSAM architecture, then PSEUDORBA cannot be used. The only known application package that has this limitation is SAP.
IAM has a feature that can potentially reduce physical I/O's for files that are randomly read. This feature, called IAM's Dynamic Tabling, offers significant performance benefits for some applications. With this feature activated, IAM will table records retrieved randomly from a file in virtual storage, without any programming changes to existing applications. Then, on subsequent random reads, IAM checks to see if the requested record is contained in the table. If the record exists in the table, IAM passes it back to the user, eliminating the I/O to the disk. If the record does not exist in the table it will be read and the record will be moved into the table. Records that are updated will be updated both in the table and on DASD.

IAM Dynamic Tabling virtual storage can come from either the users address space or from an Z/OS data space. The Dynamic Data Space feature will allow for more storage to be devoted to the table, along with improved record and storage management within the table.

NOTE: Random reads which are eligible to use the dynamic tabling feature are identified in the IAMINFO report as GET RANDOM commands. Other types of retrievals cannot use the dynamic table because the precise key being sought is unknown.

To enable this option, use the IAM ACCESS Override statement. The keywords on the override control statement are DYNDS= or DYNCORE= (See Section 30). The DYNDS= is specified when the storage is to come from a data space, and is specified in megabytes with values up to 2048. The DYNCORE value is specified in 1024 byte (1K) increments, and the storage will be acquired within the users address space. The example below is an example of using 1 gigabyte (1024 megabytes) of storage in a data space:

```
ACCESS DD=iamfile,DYNDS=1024
```

The following example will reserve 200K of storage in extended private (AMODE31) for the Dynamic Table:

```
ACCESS DD=iamfile,DYNCORE=200
```

Applications which will benefit the most from Dynamic Tabling are those with high file activity where a subset of records in the file are repeatedly being read, with few ever updated. Small files with high random activity and few updates become in core tables without the need for any programming changes.

IAM's run time IAMINFO report reflects the way an application uses the file. The report includes statistics on requests processed, I/Os to disk, Dynamic Table usage and the number of records retrieved from the Dynamic Table.

**DYNAMIC TABLE RETRIEVALS** - displays the number of record requests satisfied from IAM's Dynamic Table.

**DYNAMIC TABLE RECORDS** - displays the maximum number of records in the table.

**NOTE:** IAM Dynamic Tabling facility will not be used for files that contain spanned records, due to the very large amount of storage that would be required.
**Mass Sequential Deletes**
IAM has incorporated an ability to temporarily logically delete a record from a data block. Then, when that data block is about to be written out to DASD, the records are physically deleted from the data block. This eliminates the overhead of constantly moving records within a data block, as prior records are deleted, only to end up being deleted as well. In the case where every single record is deleted from a data block, this enhancement eliminates the data movement entirely from that process.

**Long Key Compression**
IAM files that have an index that exceeds 8000 bytes in size are automatically compressed to reduce index storage. Long keys however experienced less benefit due to the compression technique used. This is resolved with the use of the IAM Long Key Compression function, which now defaults to being used for keys that are 33 bytes or longer. This key compression enhancement may provide a significant reduction in virtual storage, particularly for very large IAM datasets with long keys. For example, a customer had a 33-gigabyte file that had a key length of over 128 bytes. The resulting prime index prior to long key compression required 227 megabytes. With the long key compression, the index size was reduced to 49 megabytes, a 78% reduction in virtual storage requirements!

The IAM Global Option `LONGKEYS=` specifies the minimum key length to be eligible for long key compression. The default value is 33, which has been reduced from the prior 128.

**Storage Pooling**
IAM has implemented an alternative storage management technique for handling below the line storage. IAM typically uses a minimum of 4K below the line storage for each opened Enhanced Format file. The storage is required for various I/O control blocks, such as the DCB, IOB’s, channel programs, and JFCB, which still have to be kept below the line. The management of this storage has been divided into three different storage pools. There is a pool for the base required storage for each opened IAM file, containing the DCB, JFCB, and related storage areas. There is a separate pool for the IOB and basic channel program, and a third pool for the sequential chained channel programs. The IOB’s and chained channel program pool entries are acquired and held only for the duration of physical I/O’s, which provides the reduction in the overall long term storage usage because they are now shared at the address space level rather than on a file by file basis.

It is anticipated that the below the line virtual storage usage will be reduced by at least 50% per opened Enhanced Format file, when several files are opened. The use of this feature is controlled by using the IAM Global Option `BELOWPOOL=YES`, which is the current default value. Although it is not recommended, users can turn this option off if necessary.

**Dynamic Region Size Adjustment**
As customers have converted files to IAM, they occasionally hit the Z/OS default limit of 32 megabytes extended private storage or their REGION size limit. This has in the past necessitated the modification, or in many circumstances writing and supplying an IEFUSI exit. Innovation has developed, and distributed a sample IEFUSI exit, that frequently can be used with minor modifications. This exit is distributed in the ICL (Installation Control Library) provided with IAM.
Because many customers have unexpectedly hit this limitation, IAM has the ability to dynamically increase the extended private region limit. This may be invoked during a file open, or when storage acquisition for a critical storage area fails. When invoked IAM will then attempt to increase the extended private storage limit by the quantity required rounded to 4 megabytes. By default, IAM will not increase the extended private region to greater than 512 megabytes. This maximum value can be either increased or decreased through the IAM Override facility, or by changing the IAM default in the IAM Global Options Table.

Dynamic region adjustment affects only normal file access; it does not function during a file load process. The value can be changed in the IAM Global Options Table, by using the keyword MAXREGION. Refer to Section 91 for information on changing IAM Global Options. The region adjustment can also be controlled with an IAM Override. The IAM ACCESS override keyword is MAXREGION, and it specifies the upper limit for the above the line storage, in megabytes. Note that by specifying the keyword DD=&ALLDD, the value will be effective for all IAM files accessed in that job step unless there is a specific override.

IAM will recognize that an ACB has already been opened for a file, within the same task, and share the index structure and buffers. For applications, or CICS regions, where a file was opened under one DDNAME or ACB for read only, and the other for UPDATE, they occasionally experienced an inability to retrieve an updated data record. This will no longer be the case. This support also reduces the storage requirements for having multiple ACB’s opened to the same file.

A special feature is available when the opened ACB is read only, and an ACB is subsequently being opened for update in the same address space. When that occurs, the overflow index is completely rebuilt, and all the buffers refreshed so that any updates that may have been done by batch jobs will be automatically and immediately available once the update ACB has completed the open process. For online systems that utilize two ACB’s, where one is read only and the other update, the update ACB can be closed to allow batch updating. The read only ACB can remain open for processing, however it may not have access to all of the records updated by the batch job(s) unless it is using IAM/RLS or IAM/PLEX. When the batch updating terminates, the update ACB can be reopened, and access to all of the batch updates will be immediately available.

IAM provides support for the Backup While Open (BWO) service interface. IAM will keep the BWO information, consisting of status flags and a time stamp. The BWO callable service will function for IAM files as it does for VSAM files, to retrieve and update the BWO data. Additionally, when BWO data is present in the IAM file control information, that information will be displayed in the IAMPRINT LISTCAT report. This will primarily be of benefit for recovery programs, such as CICS/VR, which will utilize the BWO information.
3 IAM DATA SET STRUCTURE

IAM Data Set Structure Overview

IAM provides improved levels of performance, efficiency and reliability unsurpassed by any other index file processor. To gain these advantages, rather than attempting to manage a VSAM structure more expertly, IAM establishes its own uniquely structured data set.

The organization of an IAM data set is structurally simpler than its VSAM equivalent. An IAM data set is a relative block non-VSAM (DSORG=PS) organized and managed by IAM using the EXCP access technique.

Complementing overflow structures within an IAM file can accommodate any type of file growth. A percentage of every block in an IAM file can be left free at load time to accommodate randomly distributed additions. Free space can be reserved during file load to accommodate file expansion. Additional extents are taken as needed for file expansion.

Programs that create IAM files are device independent. IAM automatically determines the target device type and, using the file's record length and CISIZE, calculates the best block size for that device. Programs that process IAM files are totally insulated from the blocking and structure of an IAM file.

IAM files can be loaded into a new allocation or an existing file allocation with a disposition of OLD or SHR as appropriate. IAM files are always considered to be reusable, unless the IAM Global Option ENABLE=NOREUSE is set, and the file is defined with the NOREUS attribute. The exception to this is that once an alternate index is defined for a base cluster, the base cluster becomes not reusable to retain full VSAM compatibility and because there is no mechanism to update the alternate index during a file load.

IAM files are simple non-VSAM space allocations managed via the EXCP access technique. IAM data sets contain fixed length blocks, within which IAM manages the data, index, and file description information. IAM data sets can be defined across multiple volumes. They make full use of secondary space allocation and do not require contiguous (CONTIG) extents. Space for an IAM file can be defined in records, tracks or cylinders.

VSAM's 4.3 Gigabyte (GB) file size limitation on non-SMS VSAM files does not apply to IAM files.
IAM File Integrity

IAM files have been designed to provide the highest level of both file and data integrity. Some of the design factors that enhance file integrity include a prime index structure which will never be updated, minimizing as much as possible the need to update other control information about the file structure, designing a structure of adding to the existing structure, as opposed to modifying the existing file structure. The dynamically extended portion of the file structure is validated during open, with efforts made to recover from most anticipated type of structure errors that may occur. Every effort has been made to provide an expandable, non-destructible file structure that provides an outstanding level of performance and responsiveness.

For circumstances where IAM files need to be shared between address spaces, and even between LPAR’s, IAM provides IAM/RLS and the optional IAM/PLEX feature. These features provide enhanced file integrity when shared access and update processing is required along with journaling and recovery capabilities.
Enhanced Format File Structure

The IAM Enhanced File Structure has been designed to provide a base for the future enhancements and capabilities. The advantages of the Enhanced File Structure over the Compatible File structure include:

- Ability to take secondary extents after the file is loaded. The size and management of the overflow and prime extension areas are dynamically managed by IAM.
- Variable Overflow that manages space within overflow blocks as containing true variable length records. This is unlike the Compatible file format which only stores in each overflow block the number of maximum length records that will fit.
- Significantly larger number of buffers per file can be allocated, offering greater I/O savings.
- Alternate Index Support
- Sysplex Record Level Sharing Support
- Single System Record Level Sharing Support
- Support of IBM hardware compression instruction
- Support of large devices (greater than 64K tracks per volume), which includes allowing up to 123 extents per volume
- Support for EAV volumes (greater than 64k cylinders per volume)
- Support for Parallel Access Volumes

All records contained within the IAM Enhanced File Structure are treated internally as variable length records. Additionally, all files that are defined as 75 tracks or larger will be automatically eligible for the IAM data compression, which helps reduce DASD space and provides I/O performance benefits.

IAM files are self-defined, non-VSAM files. This means that the information about the file structure is contained within the file itself, rather than in the system catalog and VVDS, as VSAM files are. The information kept within these system control areas, the system catalog, the VTOC, and the VVDS for DFSMS managed volumes is the same as for any other non-VSAM file. IAM files are treated as non-VSAM files by most DASD management products, including the FDR family of products.

An IAM file, as it appears on DASD, consists of a set of unique areas. Each area will be explained in further detail below. The basic structure is:

1. File Definition and Structure Data (Two blocks)
2. Prime Data Area
3. Index to Prime Data Area
4. Hardware Compression Dictionary (if applicable)
5. Extended File Area
6. Extended area index and contents description
7. Extended Overflow and/or Extended PE blocks
The first block is initialized when an IAM file is defined with the basic file definition information, such as record length, key size, key offset, and so forth. With the IAM Alternate Index support, the information on related data sets (alternate indexes and paths) is also kept in the first block. After a file has been successfully loaded, the last I/O done to the file is to store the file structure data in the first block. Included is an indication that the file has been successfully loaded, and of course information about where the index begins, size of index, plus additional required data. The only updates done to this block after a file load are for statistical information that is written during file close processing, including information on number of inserts, deletes, and updates. The bulk of the information that is presented on the IAM Listcat report is taken from this control block. The statistical information and information about the extended file area that are kept in the first block may not be accurate, and do not have to be accurate as they are for informational purposes only. A listcat that is done while the file is open to any application program will not reflect the exact status of the file, in much the same way as a listcat on an open VSAM file will also not reflect the exact status and statistics for the file. The statistical information can become inaccurate if a system or address space fails while the file is opened for update processing, and has not been successfully closed.

The second block contains information about the extended file area, including maximum size, and the location, by relative block, of where the information about the extended area is stored. This block is typically read during open processing, and is updated immediately after a new DASD extent has been acquired, and during close processing. If the file has not been successfully closed, then open processing will detect any inconsistencies, and update this data with the correct information.

The Prime Data Area is built as the file is being loaded. This area contains the data records that were passed to IAM during the file load process. There is included some embedded free space within each block, called the Integrated Overflow area. This area is similar in concept to the VSAM CI free space, and the size is indicated by the CI% Free space parameter on the IDCAMS file DEFINE. Every loaded IAM file has a Prime Data Area, except for files that have been loaded with a single record.

This index is built during close processing for a file load. This index consists of the high key in each prime data block, and may be in a compressed format. Once a file is loaded, this index structure is never changed, until the file is reorganized or reloaded.

For datasets that are loaded using hardware compression, a copy of the compression dictionary is stored in the IAM file after the prime index. This is done even for dictionaries that already exist to insure that IAM will be able to decompress the data in the file, even if the user subsequently changes the dictionary.

The Extended File Area consists of the data that is added to the file, either as inserted records or from updates that increased the size of data records, which could not fit into the Prime Data Area of the file. This area consists of blocks containing control information about the extended file area, and extended data blocks, which can be either Extended Overflow, or Prime Extension (PE). The control information for the Extended File area is based on the internal logical structure of the file, and is not necessarily tied to actual DASD extents. This way, IAM files can have their extents merged and eliminated by products such as FDR/CPK without impacting the integrity of the IAM file structure or the data it contains.
Dynamic File Expansion

The Extended File Area is acquired and formatted as needed. When there is a need to acquire an additional data block, for either Overflow or PE, a segment of the allocated and unused space is formatted. Normally, for batch processing, up to one cylinder will be formatted with empty blocks. For online processing, normally only one track will be preformatted at a time, to reduce the impact on response time. Blocks are then assigned as needed to either Extended Overflow, or PE. Once all of the allocated space is used, additional DASD space will be requested through the normal Z/OS EOV service.

If an error condition occurs during the EOV processing, such as an X37 abend condition, it is captured by the IAM DCB ABEND exit to prevent the job from actually abending. The request requiring the additional DASD space is failed, with a file full logical error. The avoidance of the abend is done to be compatible with VSAM, which will not abend either. The user will see the IBM error messages relating to the error condition encountered and an IAMW13 File full error message.

When a new extent is acquired, additional Extended Area control blocks are formatted and written as necessary, and up to one track of empty blocks will also be formatted. The control information for the Extended File area consists of identifying the blocks that are assigned to Extended Overflow, and the blocks that are assigned to PE. Additionally, the high key for each assigned PE block is retained as the index for the PE area.

Extended PE Blocks

As records are being added to the logical end of the file, defined as having keys higher than what have been previously loaded or added to the data set, PE blocks are assigned from the Extended File Area. Once a PE block is considered full, then the high key in that block is used as the index entry. Just as with the Prime Data blocks, Integrated Free space, or CI% free space, is left in each block. This will allow for records increasing in size, as well as for later record insertions. Once a block is assigned as PE, it will remain as a PE block until the file is reorganized or reloaded. A PE block is only able to hold records that fall into the established index key range. If records are deleted from the PE block, the free space is available for expansion of existing records within that block, or for new records added to the file within the established key range.

Extended Overflow Blocks

The IAM Extended Overflow area is a record based overflow area. Extended Overflow blocks are used to handle records that are being inserted within the file when there is insufficient space within the Prime Data or Prime Extension block that the record would have been assigned to, based on the established index. Extended Overflow space will also be used when an updated data record increases in size, and there is insufficient room within the block that it currently resides in for the larger record. The format of the data in the Extended Overflow block is identical to the data in the prime blocks. With Variable Overflow enabled, IAM will fully utilize the space available within each block. When a record is deleted from Extended Overflow, the space it occupied is immediately available for reuse by ANY inserted or updated record, regardless of the key value. This eliminates the unusable lost space condition that can occur within VSAM files taking CI/CA splits, as records are deleted from certain key ranges, and new records are added in different key ranges.
The index to the Extended Overflow area is record based, that is each record in Extended Overflow has an entry in the index, consisting of the key and the block number of the Extended Overflow block containing the record. There will be an index to overflow for each of the prime data or extended PE blocks that have associated records in overflow. This enables IAM to use a compressed key structure for the overflow area, as well as potential reductions in CPU time to build and manage the overflow index when there are a very large number of records in the overflow area. The index is built when the file is opened, by reading all of the used Extended Overflow blocks, as indicated by the control information.
Prime Related Overflow

Prime Related Overflow, or PRO, is an alternative structure for the Extended Overflow area in an IAM Enhanced Format file. The PRO format is a block based overflow structure that is tied into the prime key index. This reduces the size of the overflow index because only one entry in the index per Extended Overflow block that is generated when the Extended Overflow block is first used. Therefore, all of the records within any extended overflow block had to have a relationship with the index structure to a prime block, hence the terminology of Prime Related Overflow.

PRO is designed for datasets that have a very high volume of insert activity. It is ideally suited for datasets that have insert activity that is approaching a million records or more. The advantages of PRO format files include:

1. Reduced virtual storage requirements for the index
2. Reduction in frequency of file reorganizations
3. Reduced I/O to open files with large overflow areas
4. Potential for better sequential read performance

Due to the nature of being a block oriented overflow structure will mean that such files will likely require more DASD space than a record oriented overflow and additional I/O when inserting records. PRO files will also need to be used with IAM/RLS or IAM/PLEX to provide read integrity while being updated by another job due to the potential movement of existing records.

For most IAM files, the standard record based overflow area will still provide the best performance. However for the few files with an exceptionally large volume of records in overflow, PRO will be the better choice.

When a prime block, or a related extended overflow block, has insufficient space remaining to accommodate either a record being inserted, or an updated record that has become longer, an additional extended overflow block is acquired. The original block will be divided, with some of its existing records moved to the new overflow block if necessary. The point of division will be determined by IAM based on the amount of space needed, the logical location of the new or updated record within the block, and the manner in which inserts or updates are occurring. The records moved, if any, will be those with the highest keys in the original block. The new or updated record will be placed in either the original block, or the new overflow block based on space, logical position, and processing patterns.

An index entry will be created for the new overflow block, and inserted into the index for the dataset in storage, and on DASD. This is another difference in the PRO structure, which is that there will be an index entry for the overflow blocks written out to the actual dataset itself.

A PRO overflow block will be available for reuse by either the same or a different prime block when all of the records in it have been deleted. The index entry for it will be nullified when the block is emptied out, and then reused if an overflow block is needed again, even for a different prime block. This will avoid having wasted overflow blocks when there is heavy insert and deletion activity against a dataset.
Sharing PRO Files

The use of the PRO format is recommended to be used with IAM Version 9.0 and above, and only for datasets defined with Share Option 1, or that are used with IAM/RLS or IAM/PLEX. Due to the potential of multiple records being moved on an update or insert, using the file as Share Option 2 with multiple concurrent jobs reading it, and one job updating it, will be more likely to have read integrity issues than without PRO if they do not access and update the dataset with some type of record sharing software product, such as IAM/RLS. Definitely do not use PRO files without IAM/RLS for Share Option 3 or 4 processing, as it is quite likely that the dataset integrity will be compromised.
10 IAM VERSION 9.2 USERS GUIDE

10.00 IAM Users Guide Introduction

Overview

Welcome to the IAM Version 9.2 Users Guide. This guide is designed to explain how to use IAM Enhanced Format datasets. It is primarily a task oriented guide. For each task, general guidance and reference information is provided, along with numerous examples to aid in the understanding and use of IAM.

IAM Version 9.2 utilizes various functions and features only available in z/OS operating systems and instructions available only on processors that support z/Architecture. The minimum requirements for running IAM Version 9.2 are z/OS operating system and requires a processor that supports the z/Architecture Principle of Operations -04 level (fifth edition) or above. This applies to IBM processors z/890 and z990 and above.

IAM support on systems that do not meet those requirements and are using z/Architecture machine can use up through Version 9.1. Support is provided for systems without z/Architecture support by IAM Version 8.1.

This guide assumes that the reader has a general understanding of VSAM and the related terminology, has a working knowledge of z/OS JCL, and the Access Method Services (IDCAMS) utility program. Additionally, the reader should be familiar with the IAM Features and Capabilities as described previously in this manual. The primary focus of this Users Guide is on how to, as opposed to explaining each IAM or VSAM feature and concept.

VSAM Compatibility

IAM is a high performance indexed access method, providing random and sequential access to user data with minimal computer resources. IAM provides an application program interface that is compatible to the z/OS VSAM access method, supporting the most commonly used features and capabilities. IAM can be used in place of VSAM KSDS files, which are processed sequentially or randomly by key, and VSAM ESDS files, that are accessed sequentially or randomly by relative byte address (RBA), or by control interval. IAM files can be used in place of VSAM KSDS or ESDS files that utilize the IAM supported functions without modification. IAM, with the optional alternate index feature, can be used for any alternate indexes related to VSAM KSDS or ESDS types of file that have been converted to IAM. Customers with the optional alternate index feature can also use IAM in place of VSAM RRDS (numbered) data sets. IAM provides support for both fixed length record and variable length record RRDS files. While IAM does not use the VSAM LSR buffer pool, IAM can be used by applications that indicate usage of the VSAM LSR buffering, including Batch LSR and CICS. IAM files can be processed by system utility programs, including IDCAMS and any of the many SORT software products. IAM provides support for the VSAM exit routines, as specified in the ACB EXLST, including the SYNAD, LERAD, EODAD, JRNAD, and UPAD exit types.

IAM does not support accessing KSDS type of files by VSAM relative byte addressing (RBA) or by control interval processing (VSAM RPL OPTCD=ADR or OPTCD=CNV). IAM does not support control interval updates for ESDS type of files, although control interval reads and file load processing by control interval are supported.
IAM Users Guide Introduction

IAM files are processed in a manner that is identical to VSAM. First, IAM files must be defined. IAM files can be defined with the IDCAMS DEFINE utility, through z/OS JCL VSAM allocation, or by the IAM ISPF panels. Once defined, IAM files must be initialized with user data. IAM files can be loaded with application programs, IDCAMS REPRO, or with system SORT output (SORTOUT). After a successful file load, IAM files can then be processed and updated. Alternate indexes and paths can be defined and built using IDCAMS. After significant update activity, an IAM file may need to be reorganized, just as VSAM KSDS files. The frequency of file reorganization of IAM files may be either less or more than what was required with VSAM. Innovation offers a file reorganization product, called FDRREORG, which can automate the file reorganization process for IAM files, as well as VSAM and PDS type of files.

The subsequent sections will present information and examples on how to perform all of those tasks, as well as other dataset management tasks. Before getting into the specific tasks, some general JCL guidelines are presented. While there are usually no JCL changes required to utilize IAM files, there are some additional IAM unique JCL statements that, when used, can enhance the overall IAM file usage. Such JCL changes provide for overriding various IAM default options, can provide IAM unique reports, and IAM tracing and debugging capabilities.
10.10 JCL Considerations

As a general rule, there are no JCL changes required to use IAM files. JCL parameters for VSAM, such as the VSAM AMP parameters can remain without any need for change. IAM will ignore any value for BUFNI. IAM will honor the STRNO parameter, which specifies the number of place holders. IAM will use the value specified by the BUFND parameter for the maximum number of buffers, MAXBUFNO, unless it is less than the default maximum buffers, or a MAXBUFNO override has been specified. Files with JCL for Batch LSR (SUBSYS=(BLSR,DDNAME=ddname)) can remain so specified, although IAM will not use the LSR buffer pool.

Unique IAM DD Cards

There are some unique IAM DD statements, which are described below. The use of the below mentioned DD names is generally optional, however they may be requested by IAM Technical Support for gathering additional data for problem diagnosis.

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAMDEBUG</td>
<td>When specified as a DD DUMMY, will result in IAM issuing a U0184 ABEND for various error conditions. This is done to provide a simple mechanism to obtain a dump for problem diagnosis. This DD statement will also cause various IAM error messages to appear, which normally would not appear because of commonly encountered error conditions that are typically handled by return codes. The IAMDEBUG DD statement should only be used when requested by IAM Technical Support. Using it otherwise may result in normally running jobs to abend unexpectedly.</td>
</tr>
<tr>
<td>IAMDUMP</td>
<td>An optional DD card, that when specified, will be used for dumping the IAM index data space when an IAM file is being closed under a task that is abending. This will normally be a SYSOUT dataset.</td>
</tr>
<tr>
<td>IAMINFO</td>
<td>Identifies a sequential output dataset, which is normally SYSOUT. IAM will set the required DCB characteristics for this dataset, so specification of a DCB is not recommended. IAM produces the IAMINFO reports with a RECFM=FBA and an LRECL=121. This dataset will contain a report that is produced by IAM every time an IAM file is closed. The report contains file description information, along with statistics on file usage and resource requirements to process the dataset. Use of this feature is highly recommended, and may be requested by IAM Technical Support for problem resolution, particularly for performance related questions. These IAMINFO reports can also be generated by the IAMSMF utility program provided the installation is collecting and saving the optional IAM SMF records.</td>
</tr>
<tr>
<td>IAMNOLIC</td>
<td>When specified as a DD DUMMY on job steps that perform an IDCAMS LISTCAT requests, the DD card will prevent the production of the IAMPRINT reports for IAM files. This is primarily intended for use when a job is doing a LISTCAT of an entire catalog, and it is desirable to eliminate the overhead of IAM processing, which can be significant when thousands of datasets are being processed.</td>
</tr>
</tbody>
</table>
**DDNAME**  **Description**

IAMOVRID  Specifies a card image dataset, which contains the control cards and keywords to use or modify the use of various IAM features. This is normally a DD * (SYSIN type) of dataset, however it can also be a sequential dataset, or a member of a partitioned dataset. Refer to the section on IAM Overrides for complete information on using this facility.

IAMPRINT  Identifies a sequential output dataset, normally SYSOUT that contains a report on IAM file characteristics whenever a LISTCAT ALL is done for an IAM file. IAM will normally dynamically allocate this file to the default SYSOUT class (SYSOUT=*). For TSO users, the output is automatically routed to the user's terminal. The DD card can be optionally provided by the user, to capture the output to a dataset, or route the output to a different SYSOUT class. The user need not provide any DCB information, however they should be aware that the file will have an LRECL=121 and a RECFM=FBA.

IAMWKDD  An optional DD statement that provides a temporary dataset to be used during an IAM file load to hold the index. Normally, IAM will default to using a Data Space of up to 2048 megabytes that is adequate for all except the extremely large dataset. For example datasets that are using 1/4 track for a block size, with a 64 byte key, the data space is sufficient for up to 102 gigabytes of compressed data. To force the use of the work file, the CREATE Override DATASPACE=0 must be specified. No DCB characteristics should be specified, and adequate space must be allocated.

$NOVIF  When specified within a job step, indicates that IAM will not flag dynamically allocated IAM datasets as DSORG=VS in the JFCB for the dataset. This is intended for use when running dataset management utility software, such as CA-DISK and CA-DMS.
Defining IAM Files

Overview

Before using an IAM dataset, it must be defined. This define process is identical to what is required for VSAM datasets. During the define process, IAM allocates the DASD space for the dataset, catalogs the dataset, and stores the file attributes within the dataset itself. IAM datasets can be defined by using IDCAMS, through JCL DD cards, or using a variety of methods under TSO, including through the IAM ISPF panels. Many other software products that are used to define VSAM datasets will generally also be able to define IAM datasets. This section will provide information on parameters and examples to define IAM datasets with IDCAMS, through JCL, and under TSO.

How to IAM a Dataset

For a dataset to become IAM instead of VSAM, an indication must be provided on the file definition indicating that the file is to be an IAM file. The ways to indicate this are:

1. Add the parameter OWNER($IAM) to the IDCAMS DEFINE command, or
2. Change the dataset name to include the literal $IAM somewhere within the name, or
3. For SMS managed IAM datasets, use an SMS Data Class or Storage Class with the literal $IAM as part of the class name.

For most datasets, all that is required to implement IAM is to change the file definition using one of the above techniques. Any of the above methods can be specified on an IDCAMS DEFINE. For JCL definition of an IAM dataset, the OWNER parameter is not available, so either the dataset name has to include $IAM or it must be a part of the assigned Data Class or Storage Class name. Most installations select one method as their preferred method, based upon their internal dataset management or accounting requirements. While many installations have chosen the OWNER($IAM) technique, the alternative of placing $IAM within the dataset name has the advantage of making identification of IAM datasets very easy and many installations have chosen this route as well.

Basic Define Parameters

For any IAM dataset, there is certain basic information that must be provided. This is usually provided through keywords specified either on the IDCAMS DEFINE command, or as JCL. Other sources of these attributes are an SMS Data Class, or from another IAM or VSAM dataset as a model. The basic required information for all types of IAM datasets include:

- Dataset Name
- Indication that file is to be an IAM file, e.g. OWNER($IAM)
- Volume(s) on which dataset is to reside
- Quantity of DASD space required
- Maximum record size
- Type of dataset (i.e., KSDS, ESDS, RRDS, AIX, or PATH)
- Key length and offset (RKP) for KSDS (INDEXED) and AIX type of files
Additional information that can be provided includes free space, share options, and expiration date. Several of the other VSAM file attributes can be specified, such as SPEED or RECOVERY however they are not relevant to an IAM dataset and will be ignored. Certain attributes unique to IAM can be specified via IAM overrides, which include data compression, IAM file format, and default buffering range. The various unique IAM attributes can also be set as installation defaults in the IAM Global Options Table.

If the IAM allocation encounters any errors, the error messages will appear on the JES job log, with the z/OS allocation messages (SYSMSGS) and on the IDCAMS SYSPRINT, if it is available. Due to the manner in which IDCAMS prints messages on SYSPRINT, the error messages from IAM will precede the DEFINE command. IDCAMS will also print out additional error messages after the DEFINE, performing an analysis on the return codes set by IAM. Whenever possible IAM uses the VSAM return codes that most clearly indicate the actual problem, although that is not always possible. Always refer to the IAM and related allocation error messages for the most precise problem determination possible. Refer to Section 80.20 for the meaning of the return codes and reason codes returned from IAM after a DEFINE has been processed.
Defining IAM Files

Considerations for Defining an IAM Dataset

In general one can easily convert a VSAM cluster to IAM just by modifying the DEFINE, as described above. Because IAM has a different file structure, and is allocated as a non-VSAM dataset, there are some differences between IAM dataset allocations and VSAM allocations that may affect a few of your datasets.

**Data Set Type**

IAM datasets are stored as z/OS non-VSAM type of datasets. They are stored in a DSORG=PS type of file, and can be basic file type, a Large Format Sequential dataset, or an SMS Extended Format dataset. The latter two types are necessary if the file will need to use more than 64K (65,535) tracks on any single volume. *Except when absolutely necessary, Innovation does not recommend the use of SMS Extended Format datasets because they have a 32-byte suffix on each block which reduces the maximum amount of data stored per track, and require that IAM use BSAM rather than EXCP for certain processes which can result in less efficient I/O.*

**EAV Support**

IAM Version 9.0 and above provides support for IAM datasets to reside on EAV volumes. To make use of EAV’s, IAM datasets must either specify the EATTR(OPT) parameter on the DEFINE CLUSTER, be assigned to a Data Class with the EATTR(OPT) parameter specified, or have the IAM Global Option EAV enabled.

*To use IAM datasets on EAV volumes, users must be running z/OS 1.12 or higher. Users running z/OS 1.11 can use EAV volumes however the dataset must be an SMS Extended Format.*

**Selecting Data Set Type**

IAM supports an IAM CREATE Override of DSNTYPE=. Valid values are BASIC, LARGE, or EXT. Specification of this override will supersede any dataset type selected within the Data Class. The use of this new override does require users to be running z/OS 1.7 or above.

BASIC indicates the use of a standard non-VSAM type of dataset, LARGE indicates to make the dataset a Large Format Sequential dataset, and EXT indicates to use SMS Extended Format.

*To use more than 64K tracks per volumes, the IAM dataset must be either a Large Format Sequential dataset running under z/OS 1.7 or higher, or it must be a SMS Extended Format dataset. Otherwise, the amount of space that an IAM dataset can use per volume is limited by z/OS to 4,369 cylinders, which is equal to 65,535 tracks.*
Using Extended Format Datasets

IAM datasets that are SMS Extended Format datasets can have up to 123 extents per volume. To obtain an Extended Format dataset, the DFSMS data class (DATACLASS) must specify “Dataset Name Type” of “EXTENDED REQUIRED or EXTENDED PREFERRED or use the IAM CREATE Override DSNTYPE=EXT, and have a storage class (STORCLASS) with a Sustained Data Rate (SDR) of either blank or 0.

The disadvantage of using Extended Format datasets is that each block has an additional 32 bytes of system data appended to it, so such datasets may use more DASD space than when not in a SMS Extended Format. Because of this, it is recommended that customers use the Large Format Sequential datasets when possible instead of the SMS Extended Format. Setting the IAM Global Option ENABLE=LARGE is a generally safe way to automate the use of Large datasets without requiring any JCL or SMS class changes.

Because SMS appends 32 bytes of system data to each block, IAM will automatically adjust its block size calculations to consider this requirement; however the resulting dataset may require more DASD space than a non-Extended Format. IAM uses BSAM WRITE macros when an Extended Format dataset is being loaded, therefore, the EXCP counts for file loads of IAM Extended Format datasets will be the number of blocks written to DASD, not the actual number of EXCP’s issued, which will result in higher EXCP counts for file loads of these datasets. Innovation recommends that customers use Large Format datasets when possible instead of the SMS Extended format datasets.

DASD Space Consideration

IAM datasets typically require from 30 to 70% less disk space than your existing VSAM clusters. IAM datasets use DASD space more efficiently. The compressed index and advanced internal structure usually result in about a 20 to 40% reduction of disk space compared to a similar VSAM cluster. IAM’s data compression may provide an additional 20 to 50% reduction in disk space for most datasets.

Customers can consider using the IAM hardware compression instead of the software compression. IAM will automatically generate a hardware compression dictionary using the first several records that are loaded if hardware compression is indicated. Users that are interested in the best possible results from hardware compression should consider building their own customized dictionary for the desired datasets, using the IBM provided REXX EXEC. Information on using that process is described in Section 10.25.

In an effort to conserve disk space and prevent over allocation, IAM releases space that is unused and has not been reserved after the file is initially loaded. This is done automatically when secondary space is specified. If you want to override IAM’s default of releasing the over allocated space, see IAM Override statements. The keyword is RELEASE=NO. After a file has been loaded, a LISTCAT ALL will show you the exact number of tracks an IAM file is using and has allocated. Innovation recommends that when converting VSAM files to IAM, initially retain the original VSAM space allocation values. After observing the IAM space requirements, the space allocation can be adjusted if so desired.
Multivolume Consideration

IAM utilizes standard z/OS services to acquire additional DASD space. Because IAM datasets are non-VSAM, the rules and mechanisms for acquiring additional space for multi-volume datasets are different than VSAM. When IAM needs additional space, it issues the z/OS EOV (End of Volume) service to acquire additional DASD space. The only input IAM can provide is a space quantity, by specifying the desired secondary quantity in the JFCB. IAM will attempt adjustments on the secondary quantity, as per the MAXSECONDARY and MULTIVOLUME parameters. There are some circumstances where this secondary value is not honored, particularly for DFSMS managed datasets, in which case the secondary value that was indicated when the dataset was originally defined will be used.

The basic rules for IAM datasets are that the primary space quantity has to be available on the first volume. Then, as new extents are acquired, generally additional space will be acquired on the current volume, until the dataset has sixteen extents on that volume, or there is insufficient space to satisfy the request. Then z/OS will switch to the next candidate volume. When the next candidate volume is explicitly named, which is typical for datasets not managed by SMS, there must be sufficient space on the next candidate volume for the requested secondary quantity. In other words, z/OS non-VSAM EOV cannot skip over a "candidate" volume due to insufficient space and go on to the next. Attempting to do so will cause a SE37-08, resulting in an IAMW13 File full message.

There are some exceptions to the above described processing. If the file is SMS managed in a guaranteed space Storage Class then the primary allocation is made on each volume when the dataset is defined. If the secondary space is zero then once the primary allocation is used on one volume IAM will be switched to the next volume. When all the allocated space is used attempts to add more data will fail due to a file full error. When a secondary quantity is specified then additional extents will be acquired on the current volume, providing there is space to do so, until the dataset has reached either the limit of 16 extents on that volume, or has run out of space. Then IAM will be switched to the next volume. IAM compatible format datasets cannot exceed a total of 255 extents.

A second exception occurs when the IAM file is a SMS Extended Format dataset. The SMS Extended Format datasets can have up to 123 extents per volume. Therefore IAM datasets that are SMS Extended Format will not be subject to the IAM space adjustment function. The SMS Extended Format datasets are still limited to a maximum of 59 volumes.

A third exception occurs when the file is defined as multi-volume, not SMS managed and the user has specified a secondary quantity of zero. For systems that have SMS active, IAM will treat this type of allocation like an SMS guaranteed space request. The primary quantity is allocated on each volume when the file is defined. When all the allocated space is used on one volume, then the allocated space on the next volume will be used. Unfortunately, this technique does not work on systems that do not have SMS active. So, for those systems, IAM will set the secondary to be the same as the primary. This usually results in only the primary space being allocated on the first volume, and then potentially multiple extents on the second and subsequent volumes.

Reorganizing, reloading, restoring, or snap copying into a multi-volume IAM dataset without deleting and redefining is not recommended because that can result in some strange space distribution across the volumes, and may result in other problems. For restore or copy it is recommended that the utility being used perform the allocation on the target volumes.
This is because the processing done by z/OS EOV when using a previously existing file is different than for new files, and can result in secondary allocations being different than expected. This may also cause problems when restoring an IAM dataset from a backup taken prior to the reorganization.

**DFSMS Support**

IAM provides full support for SMS managed IAM datasets. By definition, to be eligible for an SMS managed volume the dataset must be assigned a Storage Class. The DATACLASS, MGMTCLASS, and STORCLASS can be explicitly provided on the DEFINE command or selected by the ACS routines. The Data Class can provide file characteristics for the file being defined including record length, key length, key offset, share options, free space, and others eliminating the need to specify those values explicitly on the DEFINE. As per SMS rules the options in the Data Class will be used unless explicitly overridden on the DEFINE command. IAM files on SMS managed volumes will be cataloged with the class names.

**DFSMS ACS Routines**

A special consideration for IAM files defined with IDCAMS in an SMS environment is that the ACS routines will be invoked twice for the define request. The first time is for the VSAM define process. If the dataset is to be an IAM dataset, then the dynamic allocation issued by IAM will result in a second call to the ACS routines as a non-VSAM dataset. The exceptions to this are when files are allocated through JCL or files being allocated through the IAM ISPF panels. In the exception cases, IAM calls DADSM directly so only the first VSAM call is performed.

When IAM issues the dynamic allocation request, IAM will specify whatever SMS classes have been assigned to the dataset at the point in time that the intercept occurred. IAM will issue the dynamic allocation request with a DDNAME of @#$IAM. This will provide a method for customers to use in their DFSMS ACS routines to identify IAM dataset allocations using the &DD read-only variable.

There are a few exceptions to using the DDNAME=@#$IAM in the allocation. These include the following circumstances:

1. For non-DFSMS allocations when the VAM option has been enabled.
2. For datasets being “defined” through JCL.
3. If the job step already has a DD card with the @#$IAM name specified.

One of the problems that can occur is if an installation has established an SMS Storage class, such as STORCLASS(NONSMS), which users can code to prevent a dataset from being SMS managed. That storage class name is nullified on the first pass through the ACS routines, causing the file to be unmanaged and preventing IAM from seeing that original storage class specification. When the ACS routines are called again out of the dynamic allocation issued by IAM, because the STORCLAS is null, the ACS routines assign the dataset to an SMS managed STORCLAS. This problem is resolved by the use of the IAM STORCLAS Global Option. By setting that option to be the non-managed storage class, (e.g. NONSMS), IAM will pass that as the STORCLAS on any allocations which do not have one at the time IAM intercepts the request.
Another problem that can occur is if the ACS routines decide on the second pass that the dataset is not to be a DFSMS managed dataset. This will cause the dynamic allocation to fail, because the volumes that are passed are DFSMS managed volumes. To prevent this from occurring, the ACS routine must not nullify that STORCLAS for an IAM file that has a STORCLAS specified. The STORCLAS can be changed to a different STORCLAS if desired.

A third consideration for the ACS routines is that some installations have set their ACS routines to perform different actions if some of the classes are already specified on entry to the routine. This is usually due to installations wanting to limit the external use of SMS classes by their users. Because of this, ACS routines with code that checks for the preexistence of SMS classes, and performing different actions, could result in IAM files being assigned to different classes than expected or desired.

The main point here is that the developer of the ACS routines, particularly the Storage Class routine, must be aware of how IAM file allocations work and must code the routines to achieve their installation's desired results with the above considerations in mind. Establishing an installation standard to use $IAM in the dataset name, or as part of a user specified DATACLAS will make it much easier to identify IAM files in the ACS routines. (Note that the OWNER parameter from the DEFINE is NOT accessible to the ACS routines.) An alternative to identifying IAM files in the ACS routines is to check for &DD being equal to '@#$IAM'. The IAM technical support team is available to help review ACS routines, and make suggestions on revising them to meet their objectives for IAM files.
For datasets that are not managed by DFSMS, IAM offers allocation to non-specific volumes. If you wish to use the IAM non-specific allocation, specify VOLUME(ANYVOL). For multi-volume non-specific allocation, specify VOLUME(ANYVOL ANYV01 ANYV02 ...). This will result in IAM issuing a non-specific dynamic allocation request for the IAM file. The first volume will be selected by z/OS allocation. Any additional volumes are selected by IAM, which will select volumes from the specified UNIT name that are of the same device type as the first volume selected. IAM builds a list of the eligible volumes and then selects those volumes that have the largest quantity of available contiguous space. All of the volumes must be mounted as STORAGE to be eligible for selection.

Volume selection is done at the time the dataset is defined, and a subsequent LISTCAT will show the volumes selected. The default UNIT name used is SYSDA. To change to a different UNIT, use the IAM CREATE override keyword UNIT=, or change the IAM Global Option WORKUNIT.

When using the IAM non-specific allocation, do NOT specify UNIQUE. Doing so will cause IDCAMS to attempt to allocate the nonexistent volumes. Similarly, if the MODEL parameter is specified, also specify the SUBALLOCATION parameter, because the MODEL parameter causes the UNIQUE attribute to be assigned.

Customers that are using the CA-ALLOCATE (formerly Sterling Software’s product SAMS also known as VAM) and that have enabled the IAM VAM Global Option cannot use the non-specific volume allocation feature of IAM. They instead should use the pooling and volume selection provided by that product.
IAM Overrides for IAM Dataset Definition

There are various unique attributes for IAM datasets that can optionally be set when the dataset is defined through the use of the IAM Override facility. The IAM override facility is provided because there is no mechanism within the DEFINE CLUSTER command to pass this information. Included among the attributes that can be specified are IAM Data Compression, a hardware compression dictionary, and secondary space options. Most of these attributes are based on defaults from the IAM Global Options Table, which are normally set up during the IAM product installation. With appropriate choices made during installation, there should be a very infrequent need to use the IAM Overrides during the file define. An overview of the IAM Override facility is presented here, while complete information is provided in Section 30 of the manual.

IAMOVRID DD Statement

The use of the IAM Override facility is triggered by providing an IAMOVRID DD card in the JCL, which usually consists of in-stream card image input (i.e. //IAMOVRID DD *). This DD can also reference a card image sequential file on DASD, or a member of a PDS. Each control card contains a word indicating the type of override (i.e., CREATE or ACCESS), then after a blank is followed by one or more operands. For a file definition, use the CREATE override statement. The CREATE override requires either the DD= or the DSN= operand, which identifies the dataset to which the override is applicable. A value of DD=&ALLDD will apply the override to any IAM dataset that is not explicitly overridden.

The IAM Override facility can be used for the Define Cluster commands issued under IDCAMS or TSO. It does not work for defines done through JCL DD cards or the TSO ALLOC command. Various IAM overrides can be specified directly on the IAM ISPF panel. The IAM Override process will most likely work with other software products that can Define VSAM files. Examples of using the IAM Override facility for file definition are included in with the IDCAMS Define Cluster examples later in this chapter.

CREATE Override Operands

The following list contains the CREATE Override keywords which are applicable to the file definition process. For the most part, the CREATE Override keywords have the same meaning and implication when used during the file load, as they do during the file define. The underscored portion of the keywords indicates their minimum abbreviation. The keywords applicable to the define process are:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLKSIZE=nn</td>
<td>Specifies the block factor (1 - 15) or block size of the IAM dataset. The default value is based on the specified CI Size and maximum record size. Typically the default is the maximum block size that can be used to obtain 4 blocks per track.</td>
</tr>
</tbody>
</table>
**Keyword**  
**BUFSPACE=**  
Specifies the maximum amount of storage in kilobytes (1024) that can be used for buffers. If specified other than 0, the value will supersede any MAXBUFNO override. This value is saved in the data characteristics and will be used to compute MAXBUFNO when the dataset is opened for processing. MAXBFNO is computed as:

\[
\frac{(BUFSPACE \times 1024)}{BLOCKSIZE}
\]

Default is no buffer space will be permanently associated with the dataset.

**DATACOMPRESS=**  
Indicates whether IAM is to data compress this dataset. Valid values are:

- **YES** – Indicates that the dataset will have its data compressed, using the default data compression technique from the IAM Global Options Table.
- **NO** – Indicates that no data compression will be used for this dataset.
- **HW** – Indicates IAM will use the z/OS hardware compression instruction to compress the data. If the DICT= override specifying a compression dictionary name is not specified, IAM will dynamically build a compression dictionary when the file is loaded, based on the data in the first several records.
- **SW** – Indicates that IAM will use its standard software compression algorithm. No dictionaries are used with this form of compression.

The default is based on the IAM Global Options Table, which as shipped is any dataset that is 75 tracks (5 cylinders) or larger is eligible for IAM data compression. The default technique is also based on the IAM Global Options Table, which is shipped as software compression.

**DDNAME=**  
Specifies the dataset to which the override is applied. When used during a DEFINE process, the value specified MUST match the value specified for the FILE parameter. If this value is set to &ALLDD, then the overrides are applicable to any IAM file that is not otherwise explicitly overridden.
<table>
<thead>
<tr>
<th><strong>Keyword</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DICTIONARY</strong></td>
<td>Specifies the four character suffix for the name of the user provided hardware compression dictionary. The dictionary must be in load module format, with the first four characters being 'IAMD'. Review the section on Hardware Compression for information on creating and naming the compression dictionary. The default is that IAM will create a dictionary during the file load process.</td>
</tr>
<tr>
<td><strong>DSN</strong></td>
<td>Specifies the name of the dataset to which the override is applied. Either DDNAME= or DSN= must be specified.</td>
</tr>
<tr>
<td><strong>DSNTYPE</strong></td>
<td>Specifies the type of dataset to be used for the IAM file. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>LARGE – Specifies that the IAM dataset will use the Large Format Sequential dataset type, which can exceed 64K tracks per volume and is not required to reside on DFSMS managed volumes.</td>
</tr>
<tr>
<td></td>
<td>EXT – Specifies that the IAM dataset will use the SMS Extended format. These datasets can exceed 64K tracks per volume and must reside on DFSMS managed volumes.</td>
</tr>
<tr>
<td></td>
<td>BASIC – The IAM dataset will be a basic non-VSAM dataset, which is limited to a maximum of 64K tracks per volume.</td>
</tr>
<tr>
<td></td>
<td>Default value is BASIC, unless the DFSMS Data Class indicates otherwise.</td>
</tr>
<tr>
<td><strong>ENHANCED</strong></td>
<td>Specifies that IAM is to create an Enhanced format IAM dataset. This type of dataset uses a dynamic overflow area, which can acquire additional DASD extents during file updates, as needed. The default is taken from the IAM Global Options Table, which is shipped as Enhanced format.</td>
</tr>
<tr>
<td><strong>INTEGRATED=nn</strong></td>
<td>Specifies CI% free space, especially useful for data compressed ESDS datasets.</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>JRNAD=</td>
<td>For Enhanced format files, specifies the IAM journaling capabilities to be used. For files not being processed by IAM/RLS or IAM/PLEX, or using IAM/RLS journaling, users must allocate and catalog a log dataset to be used for the journaling, which is required to be the name of the IAM dataset / cluster, appended with the characters “.LOG”. IAM journaling will not be active during file loads, reorganizations, or during recovery from the journal. For files being processed or journaled by IAM/RLS or IAM/PLEX, the users must specify the journal datasets to the IAM/RLS or IAM/PLEX startup procedure if they want to use journaling, and had not specified to do so when the dataset was defined or loaded. Valid values are: <strong>BOTH</strong> or <strong>ALL</strong> - The IAM log dataset will contain both before and after images. This will enable the user to perform either a forward recovery, or a backward (backout) type of recovery. <strong>BEFORE</strong> - The IAM log dataset will contain before images of updated records. This option allows backward (backout) recoveries only. <strong>AFTER</strong> - The IAM log dataset will contain after images of updated records. This option allows forward recoveries only. <strong>NONE</strong> - The IAM journaling feature will not be used for this IAM dataset. Default value is <strong>NONE</strong>.</td>
</tr>
<tr>
<td>MAXBUFNO=nnnn</td>
<td>Specifies the maximum number of buffers to be used during file access for this dataset. A maximum of 24576 can be specified.</td>
</tr>
<tr>
<td>MAXSECONDARY=nn</td>
<td>Specifies a value from 0 - 10 as a multiplication factor for the secondary space quantity, when the dataset exceeds five extents on a volume. When overridden on the DEFINE, this will be the default for both file load and file access for this dataset.</td>
</tr>
<tr>
<td>MINBUFNO=nnn</td>
<td>Specifies the minimum number of buffers to be used during file access for this dataset.</td>
</tr>
<tr>
<td><strong>Keyword</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MULTIVOLUME=</td>
<td>Specifies space quantity to request when IAM believes that the next extent will be placed on the next candidate volume. The default is specified in the IAM Global Options Table, and as shipped is MULTIVOLUME=PRIMARY.</td>
</tr>
<tr>
<td>[PRIMARY</td>
<td>SECONDARY]</td>
</tr>
<tr>
<td>PRO=[YES</td>
<td>NO]</td>
</tr>
<tr>
<td></td>
<td>YES – Use the PRO format for this dataset.</td>
</tr>
<tr>
<td></td>
<td>NO – Do not use the PRO format for this dataset.</td>
</tr>
<tr>
<td></td>
<td>Default value is based on the IAM Global Options, which is shipped as NO.</td>
</tr>
<tr>
<td>PSEUDORBA</td>
<td>For ESDS files, indicates that the file can exceed 4 gigabytes of user data. IAM generates RBA values that are different than normal VSAM values. This attribute cannot be used with ESDS files used by software that depends on the normal VSAM values.</td>
</tr>
<tr>
<td>RELEASE=[YES</td>
<td>NO]</td>
</tr>
<tr>
<td>RLSID[=nnnn]</td>
<td>When specified with no operand, indicates that this dataset is to be processed using IAMRLS, regardless of Share Options or the IAMRLS Dataset Name Tables. The operand, if specified, is the four character identifier of the preferred IAMRLS address space that this dataset is to be processed by.</td>
</tr>
<tr>
<td>UNIT=</td>
<td>Specifies the unit name to specify when allocating the IAM dataset. The default is SYSDA for non-specific volume allocations, or the generic unit type, e.g. 3390.</td>
</tr>
<tr>
<td>VARQOVERFLOW=[YES</td>
<td>NO]</td>
</tr>
</tbody>
</table>
**XESDS | NOXESDS**

For ESDS files, XESDS specifies to use an 8-byte RBA value so the ESDS can exceed 4 gigabytes of user data. The application programs must be able to handle the 8-byte RBA. This is consistent with the VSAM Extended Addressability support. NOXESDS indicates that a 4-byte RBA will be used that will limit the size to 4 gigabytes.

Default is based on IAM Global Options which as shipped is XESDS is disabled so that IAM will use a 4-byte RBA.
Defining IAM Datasets Using IDCAMS

IDCAMS Define

When the IAM VSAM Interface is active in the system, every DEFINE is analyzed by IAM. If the DEFINE contains the $IAM parameter, the file is created as an IAM dataset. If the $IAM parameter is not coded, a VSAM cluster is created.

IAM allocates a non-VSAM file on disk with a DSORG of PS. This dataset contains the index, the data records and the file characteristics all incorporated into a single dataset. IAM dataset must be cataloged in a VSAM or ICF Catalog.

The parameters specified in the DEFINE statement are applied to the IAM dataset. Because the IAM file and overflow structures are different from VSAM's, some of the parameters specified may be changed or ignored.

When the DEFINE is for an IAM dataset the following rules will apply:

1. The cluster name will be the name of the IAM dataset.
2. The data and index component names will be ignored.
3. Index component attributes are not needed and will be ignored.
4. Attribute parameters will be used or ignored as documented in the following paragraphs.

IDCAMS JCL

There are normally no JCL changes to run IDCAMS to define IAM datasets, when converted from VSAM, unless IAM is not in the Link List or is using the IAM Override facility. If IAM is not in the Link List, then a JOBLIB or STEPLIB DD card will be necessary. Because IDCAMS operates as an APF authorized program, the IAM load library referenced by the JOBLIB or STEPLIB must be explicitly APF authorized in the IEAAPFxx member of SYS1.PARMLIB. If the library is not authorized, then z/OS will not even look in the library for the IAM modules, which will result in a VSAM file being defined instead of an IAM file. The only other DD statements required are SYSIN DD for the control card input, and a SYSPRINT DD to SYSOUT. If, as in the examples provided, you do a LISTCAT ALL, then IAM will dynamically allocate an IAMPRINT DD to SYSOUT=*. If you do not want the IAMPRINT output to go to SYSOUT=*, then provide an explicit IAMPRINT DD card.

Define with REPRO

One word of caution must be given about specifying the IAM dataset on a DD card within the job step that is defining the dataset. If the file is deleted and defined within the same execution, and is defined on different volume(s) than it was on originally, then the DD card will not reflect the newly defined allocation. The volume(s) for the DD card are allocated by z/OS when the step is initiated, so if the dataset is moved to different volume(s) as a result of the define, attempting to use the DD card for I/O will result in errors. For example, if under a single execution of IDCAMS you have the following sequence of commands:

- REPRO INFILE(iamfile) OUTFILE(seqfile)
- DELETE my.iam.file
- DEFINE CLUSTER(NAME(my.iam.file) - etc..
- REPRO INFILE(seqfile) OUTFILE(iamfile)
You may experience errors when the second REPRO is done to reload the IAM file. This is particularly true if the IAM file is a DFSMS managed dataset, which will likely change the volume(s) on which the dataset resides. The circumvention for this problem is to change the second REPRO to use the OUTDATSET or ODS parameter instead of OUTFILE, and the first REPRO to use INDATASET or IDS instead of INFILE. This will result in IDCAMS dynamically allocating the dataset at the point in time the REPRO is being done, so the correct volume information will be available. Another alternative is to execute the REPRO in a subsequent job step. This problem is not unique to IAM, it can occur with VSAM as well.

Most VSAM DEFINE parameters are applicable to IAM. The following paragraphs will document the parameters that are necessary for IAM, the ones that do not apply and any differences between IAM and VSAM.

The following parameters are for the essential information. Generally, these will be explicitly provided on the DEFINE, however the information other than the name can be filled in by using a MODEL of an IAM or VSAM dataset, or from an SMS Data Class.

<table>
<thead>
<tr>
<th>Essential Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLUSTER(</td>
<td>Identifies that a VSAM type dataset is to be created. This is a required parameter for an IAM dataset.</td>
</tr>
<tr>
<td>NAME(dsname)</td>
<td>This is a required parameter for an IAM dataset. The entry name specified for the cluster will be the name of the IAM dataset. The data and index component names are ignored. If '$IAM' appears anywhere in the name the dataset will be created as an IAM dataset.</td>
</tr>
<tr>
<td>OWNERID($IAM)</td>
<td>If OWNER($IAM) is coded on the CLUSTER statement, the file will be created as an IAM dataset. When '$IAM' is not present in the cluster name, and the $IAM parameter is not otherwise specified, the file will be created as a VSAM cluster.</td>
</tr>
<tr>
<td>CYLINDERS(xx yy)</td>
<td>Required information that indicates the amount of DASD space to be allocated for the IAM dataset. The unit of allocation is based on the keyword specified. The first value provided indicates the amount of space to be allocated during the file definition process. For IAM files, the primary quantity MUST be available on the first volume, otherwise the request will fail. The second value, which is optional, indicates the amount of additional DASD space to request in case the primary quantity is insufficient. The secondary quantity will be used to acquire additional extents during file loads or reorganizations. The secondary quantity will also be used to acquire additional space for Enhanced format IAM datasets as needed to handle record updates and inserts after the file load. This parameter may be omitted if it is specified on a DFSMS Data Class that will be used for this data set.</td>
</tr>
<tr>
<td>TRACKS(xx yy)</td>
<td></td>
</tr>
<tr>
<td>RECORDS(xx yy)</td>
<td></td>
</tr>
<tr>
<td>MEGABYTES(xx yy)</td>
<td></td>
</tr>
<tr>
<td>KILOBYTES(xx yy)</td>
<td></td>
</tr>
</tbody>
</table>

If the IAM dataset being defined is a basic non-VSAM dataset it is limited to 16 extents per volume and if Compatible format to a total maximum of 255 extents. The maximum extent size is 64K-1 tracks (65,535), or 4,369 cylinders per volume, which is also the maximum amount of space that can be used by IAM on any single volume. These limits do not apply to Large Format datasets or SMS Extended Format datasets.
<table>
<thead>
<tr>
<th>Essential Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VOLUMES(volser....)</strong></td>
<td>Specifies the volume(s) on which IAM is to allocate the dataset. Due to the TIOT limitation, IAM files can reside on a maximum of 59 volumes. The volume(s) provided can be specific volumes, DFSMS non-specific volume of *, or non-DFSMS non-specific volume of ANYVOL. This parameter can be omitted for DFSMS managed data sets.</td>
</tr>
<tr>
<td><strong>KEYS(length offset)</strong></td>
<td>For KSDS files, required parameter that specifies the length of the key, and the relative position of the key within the record. The maximum key length supported by IAM is 249. For spanned files the maximum key offset is 4091. Because IAM does not data compress the bytes preceding the key, it is beneficial to place the key as close to an offset of 0 as possible. The IDCAMS default values are: 64,0.</td>
</tr>
<tr>
<td><strong>RECORDSIZE</strong>&lt;br&gt;(average,maximum)</td>
<td>The first value specifies the average record length in bytes. The second value is the maximum record length. The maximum record size supported by IAM Enhanced Format files is almost 8 megabytes. There is not a specific maximum record size supported by IAM, as the maximum size possible is dependent on several factors, including the block size being used for the file, the key length, and the key offset. The DEFINE request will fail if IAM calculates that it cannot support the specified maximum record size. Files with large record sizes that exceed the half-track block size on the type of device the file is being defined on must be defined with the SPANNED attribute. The average record length is primarily used when the define request specifies that the space is to be allocated by RECORDS. Then IAM (and VSAM) will use the average record length when converting the space value to a device specific quantity. For RRDS (numbered) data sets, if the average record length is equal to the maximum, then a fixed length record RRDS will be defined. IDCAMS default values are: average=4089 maximum=4089</td>
</tr>
<tr>
<td><strong>INDEXED</strong>&lt;br&gt;<strong>NONINDEXED</strong>&lt;br&gt;<strong>NUMBERED</strong></td>
<td>Specifies the type of dataset that is being defined. INDEXED indicates that a KSDS type of dataset is being defined, NONINDEXED indicates an ESDS type of dataset, and NUMBERED indicates an RRDS type of dataset. RRDS data sets require the optional AIX feature license. IDCAMS default is INDEXED.</td>
</tr>
</tbody>
</table>
### Optional Parameters

While the following keywords are all optional, in various situations they may be required or quite beneficial. Some of the keywords listed here cannot be used for IAM files, and are presented for that reason. For ease of reference, they are presented in alphabetical sequence.

<table>
<thead>
<tr>
<th>Optional Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BUFFERSPACE(bytes)</strong></td>
<td>Specifies the maximum amount of virtual storage in bytes to be used for buffers for this dataset. IAM will use this value to calculate the effective MAXBUFNO for accessing the dataset, providing it does not go below the default value from the IAM Global Options Table.</td>
</tr>
</tbody>
</table>
| **CONTROLINTERVALSIZE** | For VSAM, this controls the logical and physical block size on DASD for the file. VSAM restricts the size to multiples of 512 or 2048 that usually results in under utilized disk space. IAM takes the size specified and based on the device type being allocated to and the record size, calculates a larger block size for the dataset that is a proportion of the track capacity. If the block size developed would result in more blocks per track than the default block factor, which is shipped as 4 blocks on a track, IAM will increase the block size until only that many blocks will fit on a track. The block size may be further adjusted if the block cannot contain at least 4 maximum size records. The default block factor can be changed through the use of IAMZAPOP, by setting the VSAMBLOCKF parameter. The block factor or size can also be set by the B= operand of the IAM CREATE Override. For IAM ESDS files, IAM will use the specified CI SIZE in the calculations for the RBA of each record. When reorganizing an IAM ESDS, users should be careful not to change the CI SIZE, because it can result in different RBA values for the records. This may cause problems for applications that have an index into the ESDS file by RBA. IAM does store the specified CI SIZE, so that it is available information if the dataset is converted back to VSAM. EXAMPLE OF BLOCK SIZE ON A 3390:
| CISZ(4096) IAM's BLKSIZE will be 13682 - 4 blocks per track |
| CISZ(20480) IAM's BLKSIZE will be 27998 - 2 blocks per track |
| **DATACLASS(dataclass)** | For DFSMS installations, this parameter specifies the name of the SMS DATA CLASS construct, which provides the allocation attributes for the new dataset. The DATACLAS name must have been previously established by the Storage Administrator. Attributes from the DATACLAS will be used, unless otherwise explicitly specified on the DEFINE statement. If the DATACLAS name contains the literal $IAM, and the dataset is SMS managed, then the dataset will be defined as an IAM dataset. Values provided by a Data Class include type of dataset, ESDS, KSDS, or RRDS, maximum record size, key length and key offset, space allocation values, free space, share options, CI size, and volume count. |
### Optional Keyword | Description
--- | ---
**EATTR(OPT | NO)** | Specifies whether or not the dataset can reside on an EAV volume which is one that has > 64K cylinders. Specifying OPT indicate the dataset can reside on an EAV volume. Specifying NO indicates it cannot reside on an EAV volume.

The default for IAM files is NO.

**FILE(ddname)** | Optional keyword that specifies a DD name that allocates the volume(s) on which the IAM file is to be allocated.

For IAM files, the primary use of this parameter is to provide a mechanism for relating IAM overrides when they are specified on a DDNAME level, with the specific file being defined. For example, if you specify an IAM CREATE Override with a DDNAME=FILE1, then to relate those overrides to the desired file, specify FILE(FILE1) on the DEFINE CLUSTER command. With IAM, there is no need for there to actually be a DD card in the step defining the file with that DDNAME. An alternative to using the file parameter is to code the CREATE override specifying the DSN= (dataset name) parameter.

If you are defining only one file in a particular job step, or want the same override for all of the files being defined, on the IAM Create Override statement specify DD=&ALLDD, and do not provide any FILE parameter on the define.

**FREESPACE(CI%, CA%)** | For KSDS type of files or variable length record RRDS files, specifies the amount of space to be reserved for future inserts or updates when the file is being loaded.

**CI%**: Specifies the amount, as a percentage, of space to be left available in each prime block of the IAM file. This is identical to the VSAM CI free space concept. Low values of no more than 5 are recommended for data compressed files that are subject to update processing. For most other files, 0 is generally the best choice. The concept of free space within a CI or data block works best to handle files with insert activity that are uniformly done across the entire key range of the file, which occurs quite infrequently. Larger values of 20% or higher are generally discouraged as they end up wasting a lot of space resulting in increasing file size and physical I/O.

**CA%**: For Enhanced format IAM datasets, this controls how much DASD space is released at the end of a file load. Using 1/2 of the specified percentage, a target amount of DASD space to be reserved for future expansion is computed. If the amount of available DASD space within the file extent(s) is equal to or less than the amount to be reserved, then no space is released. IAM will not go after additional extents to meet the space reservation. If the amount of DASD space exceeds the reserved value, then the excess will be released.

If your installation has the DSORG=DA set, then be aware that z/OS will not allow any space to be released.

**LINEAR** | Specifies that a LINEAR type of CLUSTER that is being defined. IAM does not support this type of VSAM dataset. Specification of LINEAR for an IAM dataset will result in the DEFINE failing.
<table>
<thead>
<tr>
<th>Optional Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG(NONE</td>
<td>UNDO</td>
</tr>
<tr>
<td>MGMTCLASS (management class name)</td>
<td>For SMS installations, this parameter specifies the name of the SMS Management Class for the new dataset.</td>
</tr>
<tr>
<td>MODEL(datasetname)</td>
<td>Specifies that the attributes of the dataset being defined will be copied from an existing VSAM or IAM dataset. This capability is only relevant for basic file characteristics, such as record size, key length and offset, space allocation values, volumes, and free space values. Any IAM Overrides are NOT picked up by the MODEL parameter. Likewise, any VSAM file attributes that are ignored by IAM are not available either, such as IMBED, SPEED, REPLICATE, KEYRANGES, etc. If you are trying to define a VSAM file using an IAM file as a model, you MUST provide an OWNER parameter, with a value that does not contain $IAM.</td>
</tr>
<tr>
<td>RECATALOG</td>
<td>Is an optional keyword for existing IAM datasets to reestablish the catalog entry. Requires the user to specify the dataset name, the volume(s), and OWNER($IAM) if $IAM is not in the dataset name. Recatalog is also used after renaming one or more components of an alternate index sphere, to reset the dataset relation names to the new names.</td>
</tr>
<tr>
<td>REUSE</td>
<td>Specifies whether the file being defined can be reloaded (or reorganized) without being redefined. IAM defaults to REUSE, which is that any IAM file can be reloaded without having to be deleted and redefined. To use this feature with IDCAMS REPRO, specify the REUSE keyword. IAM does provide a Global Option, ENABLE=NOREUSE, which if set will cause IAM to honor the specification of REUSE or NOREUSE. If that Global Option has been set, then IAM will honor the NOREUSE setting just like VSAM. A few applications are dependent on NOREUSE setting.</td>
</tr>
<tr>
<td>NOREUSE</td>
<td>NOREUSE will not allow a file to be reloaded without being deleted and redefined. An exception to this is made if the program issuing the OPEN is FDRREORG, in which case it will be allowed. If any other attempt is made to do so, the OPEN will fail with a return code of 8, and the ACB error flag set to 232, or x’E8’.</td>
</tr>
<tr>
<td>Optional Keyword</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SHAREOPTIONS (cross-region,cross-system)</td>
<td>Specifies the level of protection provided by the access method to prevent or allow sharing of data within the file. The protection mechanisms include the z/OS ENQ service, and the internal IAM buffering techniques.</td>
</tr>
</tbody>
</table>

The first parameter specifies how a file can be shared in the same system (CPU/LPAR). The second parameter specifies how a file is shared between systems or LPARs.

**NOTE:** With IAM/RLS or IAM/PLEX, IAM supports automatic record level sharing for concurrent users of the IAM dataset. When IAM/RLS and IAM/PLEX are not active or not in use, IAM supports the cross-region share options with the z/OS ENQ service, the same as VSAM. IAM does not support the cross-system share options. IAM issues an ENQ with a major name (QNAME) of IAMENQ and the dataset name plus first volume serial as the minor name (RNAME). If you need to enforce ENQ protection cross-system then you must add the major name of IAMENQ to your CA-MIM or IBM’s GRS control files or whatever ENQ control product you use.

**Cross Region Share Option Values:**

1. Any number of users for read **OR** one user for update. The file's structure, data integrity, and read integrity are fully preserved.

2. Any number of users for read **AND** one user for update. The file's structure and data integrity are fully preserved. **If the file is currently opened for update, other users reading the file do not have read integrity.** They may not be able to access records that were moved or added to the overflow areas of the file, without closing and reopening the dataset.

3. Any number of users for read or update and users are responsible for integrity. Updated blocks are immediately written back out to DASD. **Use of this share option for IAM files is strongly discouraged, unless you share the dataset with IAMRLS, IAM/PLEX, or some other VSAM sharing software. Due to the nature and structure of the index to the IAM overflow area, the data integrity of IAM files is compromised by use of this share option value.**

4. Any number of users for read or update, and users are responsible for integrity. IAM will use only a single buffer, and each logical I/O request will cause the buffer to be refreshed, and subsequently rewritten if the record is updated. **Use of this share option for IAM files is strongly discouraged, unless you share the dataset with IAM/RLS, IAM/PLEX, or some other VSAM sharing software. Due to the nature and structure of the index to the IAM overflow area, the data integrity of IAM files is compromised by use of this share option value.**

**SPANNED**

IAM supports SPANNED records on Enhanced Format files, with a maximum record size of almost 8 megabytes.
### Optional Keyword

<table>
<thead>
<tr>
<th><strong>STORCLASS</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>(Storage class name)</td>
</tr>
</tbody>
</table>

For DFSMS installations, this parameter specifies the name of the DFSMS Storage Class construct. For datasets that are to be placed on DFSMS volumes, Storage Class must be either implicitly specified by the ACS routines, or explicitly specified on the DEFINE command. If the Storage Class Name contains the literal $IAM, the file will be defined as an IAM file.

### SUBALLOCATION

IAM files are always allocated as if they were unique clusters. However, it may be necessary to specify this parameter when using the IAM non-specific allocation (ANYVOL) and the MODEL parameter. SUBALLOCATION will prevent IDCAMS from allocating the volumes indicated in the VOLUME parameter when the MODEL parameter is specified.

### TO(date)

FOR(days)

Specifies the retention period for the file being defined. This parameter has the same meaning for an IAM file as a VSAM file. The expiration date is placed in the VTOC for the dataset, and in the catalog entry. The keyword PURGE must be specified on the DELETE to cause the file to be scratched.

- **TO(date)** - gives the date in the form YYYYDDD (four or two digit year and three digit Julian date), through which the IAM file defined is to be date protected.

- **FOR(days)** - gives the number of days up to 9998, through which the IAM file being defined is to be date protected. A value of 9999 results in permanent retention.

Default: Dataset is not date protected.

### UNIQUE

This parameter has no relevance for IAM files, as IAM files are always unique.

*The user should be aware that specification of this keyword results in IDCAMS allocating the specified volumes prior to issuing the actual define request. For this reason, it is recommended that this parameter not be specified for IAM files.*

If the customer is using the IAM non-specific device allocation, i.e. ANYVOL, then this parameter must not be specified.
Examples of Defining IAM Datasets with IDCAMS

The following is a set of examples demonstrating how to define IAM datasets using IDCAMS. The first example demonstrates how to convert a DEFINE for a VSAM dataset to an IAM dataset. The subsequent examples demonstrate various different ways of using IDCAMS to define IAM datasets. All of the included examples have a LISTCAT request after the define request. When a LISTCAT ALL is done for an IAM dataset, IAM will dynamically allocate an IAMPRINT DD to SYSOUT which will contain detailed information on the IAM dataset. The output from IDCAMS itself will indicate that the IAM file is a non-VSAM dataset.

Note: The JCL used for the examples throughout the User’s Guide are contained in the JCL library created when you installed IAM. Check with the person who installed IAM to see if that library is available to you.

This side by side example demonstrates how simple it is to convert a VSAM cluster definition to an IAM dataset definition. The only change required was adding the parameter OWNER($IAM) under the CLUSTER level of the Define control statement. For many VSAM KSDS or ESDS types of files, this is all that is necessary to convert the file to IAM. Starting with the next file load, all of the performance advantages and features of IAM are available for this dataset.

<table>
<thead>
<tr>
<th>Original VSAM</th>
<th>IAM DEFINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>//DEFINE EXEC PGM=IDCAMS</td>
<td>//DEFINE EXEC PGM=IDCAMS</td>
</tr>
<tr>
<td>//SYSPRINT DD SYSPUT=*</td>
<td>//SYSPRINT DD SYSPUT=*</td>
</tr>
<tr>
<td>//SYSIN DD *</td>
<td>//SYSIN DD *</td>
</tr>
<tr>
<td>DEFINE CLUSTER - (NAME(EXAMPLE1.DATASET) - VOLUMES(VOL001) - CYL(10 1) - SPEED REUSE ) - DATA( NAME(EXAMPLE.DATASET.DATA) - RECORDSIZE(200 256) - KEYS(16 0) - CISZ(4096) - FREESPACE(10 10) ) - INDEX( NAME(EXAMPLE.DATASET.INDEX)- CISZ(1024) IMBED ) )</td>
<td>DEFINE CLUSTER - (NAME(EXAMPLE1.DATASET) - VOLUMES(VOL001) - CYL(10 1) - ADD --&gt; OWNER($IAM) - ADD SPEED REUSE ) - DATA( NAME(EXAMPLE.DATASET.DATA) - RECORDSIZE(200 256) - KEYS(16 0) - CISZ(4096) - FREESPACE(10 10) ) - INDEX( NAME(EXAMPLE.DATASET.INDEX)- CISZ(1024) IMBED ) )</td>
</tr>
<tr>
<td>LISTCAT ENT(EXAMPLE1.DATASET) ALL</td>
<td>LISTCAT ENT(EXAMPLE1.DATASET) ALL</td>
</tr>
<tr>
<td>/*</td>
<td>*/</td>
</tr>
</tbody>
</table>

Figure 10-1: Example Conversion of Define from VSAM to IAM (EX1020A)
This example demonstrates a basic IAM dataset definition. Note, in comparison to the above example, that the DATA and INDEX component sections are eliminated. There is no need for them to exist for IAM dataset, as IAM consists of a single physical dataset, which uses the name specified for the CLUSTER as its dataset name. This example includes the optional parameters of FREESPACE, SHAREOPTIONS, and REUSE. While in general REUSE is not necessary, as it is the default for IAM datasets, it is specified just in case the installation has changed the IAM Global Options to ENABLE=NOREUSE, in which case the IDCAMS default of NOREUSE takes effect without the explicit specification of REUSE.

```bash
//IAMDEFIN EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *
  DEFINE CLUSTER -
    (NAME(MY.IAM.KSD) -
    OWNER($IAM) -
    VOLUMES(MYVOL1) -
    CYL(10 2) -
    RECORDSIZE(100 1000) -
    KEYS(24 8) -
    FREESPACE(5 20) -
    SHAREOPTIONS(2 3) -
    REUSE )
    LISTCAT ENT(MY.IAM.KSD) ALL
/*
```

**Figure 10-2: Example Basic IAM KSDS Definition (EX1020B)**

This example demonstrates a basic ESDS dataset definition. The file will be defined as an IAM dataset due to the literal `$IAM` included in the dataset name. Note that the keyword NONINDEXED has been included to indicate that the file is to be an ESDS type of file.

```bash
//DEFINESD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *
  DEFINE CLUSTER -
    (NAME(MY.ESDS$IAM.FILE) -
    VOLUMES(MYVOL1) -
    CYL(200 50) -
    RECORDSIZE(80 100) -
    CISZ(4096) -
    SHAREOPTIONS(2 3) -
    NONINDEXED
    LISTCAT ENT(MY.ESDS$IAM.FILE) ALL
/*
```

**Figure 10-3: Basic IAM ESDS Define (EX1020C)**
Example D: Define of an IAM ESDS for > 4 Gigabyte

Unless the IAM Global Option has been changed, IAM ESDS datasets cannot exceed 4 gigabytes and will utilize a 4-byte RBA. Below is an example of using the IAM CREATE override of XESDS to force an 8-byte RBA. Note that such files do not have to reside on SMS managed volumes. If you need a >4 gigabyte ESDS file and your program(s) do not support an 8-byte RBA, you can substitute PSEUDORBA for the XESDS override keyword.

```plaintext
//BIGESDS  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//IAMOVRID DD *  
CREATE    DD=&ALLDD,XESDS
/*
//SYSSIN DD *
DEFINE CLUSTER -  
  (NAME(MY.BIG.ESDS)-  
    OWNER($IAM) -  
    CYL(1100 1100) -  
    NONINDEXED -  
    RECORDSIZE(200 4089)-  
    CISZ(4096) -  
    VOLUMES(V33901 V33902 V33903 V33904) )  
LISTCAT ENT(MY.BIG.ESDS) ALL  
/*
```

Figure 10-4: Define of > 4 GIGABYTE IAM ESDS File (EX1020D)

Example E: Define an IAM Dataset Using a Model Dataset

In the example below, an IAM dataset is defined using a model dataset. The model dataset can be either VSAM or IAM. Amongst the attributes that will be used from the model, unless otherwise explicitly specified on the define request, are file format (KSDS or ESDS), record sizes, key length and offset, free space values, volume, and control interval size.

```plaintext
//DEFMODEL EXEC PGM=IDCAMS  
//SYSPRINT DD SYSOUT=*  
//SYSSIN DD *  
DEFINE CLUSTER -  
  (NAME(MY.NEW.IAMFILE) -  
    OWNER($IAM) -  
    MODEL(MY.INDEXED.FILE) )  
LISTCAT ENT(MY.NEW.IAMFILE)  
/*
```

Figure 10-5: Example of an IAM Define with Model (EX1020E)
In the example below, an IAM dataset is being defined that will be SMS managed. The SMS classes are explicitly specified by the user. One or more of the SMS classes could be automatically selected by the ACS routines. All of the dataset attributes are being determined by SMS from the appropriate SMS classes. The following attributes are taken from the specified Data Class:

- Type of dataset (KSDS, ESDS, or RRDS)
- Maximum Record Size (LRECL)
- Key length and offset (for KSDS datasets)
- Free Space and Share Options
- Space allocation parameters
- Control Interval Size

The dataset will become an IAM dataset because the $IAM literal is contained within the Data Class name. Note that while a Data Class can be specified for datasets not managed by SMS, for IAM to pick up the Data Class name the dataset MUST be SMS managed. Otherwise, IDCAMS does not pass the Data Class value on the DEFINE, although the other Data Class attributes are included.

```plaintext
//DEFSMSDS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
  DEFINE CLUSTER - 
    (NAME(MY.SMS.FILE) -  
     DATACLAS(MY$IAM1) -  
     MGMTCLAS(DBSTNDRD) -  
     STORCLAS(BASE) )  
     LISTCAT ENT(MY.SMS.FILE) ALL  
/*
```

Figure 10-6: Define of an SMS Managed IAM Dataset (EX1020F)
Example G: Defining Multiple IAM Datasets with Overrides

This is an example of defining multiple IAM datasets within the same IDCAMS job step. Each IAM dataset has its own unique overrides. Each of the overrides identifies the related dataset by using the DSN= (dataset name) operand. The first dataset has very high I/O activity, so it is being defined with a MAXBUFNO override to increase buffering. The second dataset is used by COBOL programs with a variable length record layout giving a maximum record size of 32,200. It is defined as containing SPANNED so that it will not use an inefficient block size of 32760. This dataset has an override to indicate that it is to use hardware compression.

```
//DEF2FILE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=* 
//IAMOVRID DD *
CREATE DSN=MY.HEAVYIO.CLUSTER,MAXBUFNO=512 
CREATE DSN=MY.BIGREC.CLUSTER,DATACOMP=HW
/*
//SYSIN DD *
DEFINE CLUSTER -
  (NAME(MY.HEAVYIO.CLUSTER) -
  OWNER($IAM) -
  RECORDSIZE(1000 2000) -
  KEYS(24 0) -
  VOLUMES(MYVOL1 MYVOL2) -
  CYL(500 100) -
  FREESPACE(10 10) -
  SHAREOPTIONS(2 3) -
  REUSE )
LISTCAT ENT(MY.HEAVYIO.CLUSTER) ALL
DEFINE CLUSTER -
  (NAME(MY.BIGREC.CLUSTER) -
  OWNER($IAM) -
  RECORDSIZE(500 32000) -
  KEYS(32 0) -
  VOLUMES(MYVOL3 MYVOL4) -
  CYL(1000 200) -
  FREESPACE(10 10) -
  SHAREOPTIONS(2 3) -
  CISZ(8192) -
  REUSE SPANNED )
LISTCAT ENT(MY.BIGREC.CLUSTER) ALL
/*
```

Figure 10-7: Multiple Dataset Define (EX1020G)
Example G1: Fixed Length Record RRDS

This example demonstrates the definition of a fixed length record RRDS (numbered) IAM dataset.

```plaintext
//IAMDEFIN EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
   DEFINE CLUSTER -
      (NAME(MY.IAM.RRDS) -
       OWNER($IAM) -
       NUMBERED -
       VOLUMES(MYVOL1) -
       CYL(10 2) -
       RECORDSIZE(100 100) -
       SHAREOPTIONS(2 3))
   LISTCAT ENT(MY.IAM.RRDS) ALL
/*
```

Figure 10-8: Example Fixed Length Record IAM RRDS Definition (EX1020G1)

Example G2: Define on an EAV

This example shows the use of the EATTR parameter to have a dataset allocated to an EAV volume.

```plaintext
//IAMDEFIN EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
   DEFINE CLUSTER -
      (NAME(MY.EAV.CLUSTER)-
       VOL(*) -
       CISZ(4096) -
       OWNER($IAM) -
       EATTR(OPT) -
       DATACLAS(VSAMX4G) -
       STORCLAS(SCEAV) -
       CYL(2800 700) -
       RECORDSIZE(500 500) -
       KEYS(50,8) -
       FREESPACE(00,00) -
       SHAREOPTIONS(2,3))
   LISTCAT ENT(MY.EAV.CLUSTER) ALL
/*
```

Figure 10-9: Example of defining on an EAV volume (EX1020G2)
Defining IAM Files in JCL

IAM datasets can be defined in JCL, just as VSAM datasets can. DFSMS must be active within the system to use this facility; however the dataset does not have to be DFSMS managed. Similar to VSAM, only a subset of the attributes can be specified in JCL. Use of a DATACLAS is encouraged, because that will pick up the free space and share option values, which cannot be specified in JCL. Also, because the JCL defines are actually done under the initiator TCB, IAM does not have access to any IAM Overrides that may be included in the JCL for the define process. The overrides can be specified on the job step that loads the dataset, which may or may not be the step that allocates the IAM dataset. Because there is no mechanism to pass an OWNER parameter, the dataset must either have the literal "$IAM" within the dataset name, or be a DFSMS managed dataset with "$IAM" in either the STORCLAS or DATACLAS name.

IAM files can also be allocated as temporary datasets, with the same restrictions as temporary VSAM datasets. The dataset will not be cataloged, and it is restricted to a single volume.

To use the IAM JCL define capability, IAM MUST BE IN LINKLIST! Because the definition is running under the initiator TCB, there is no access to the STEPLIB or JOBLIB to do the allocation. If you have multiple levels of the IAM active, then the level in the link list will perform the define request.
The following is a list, with a brief description of the JCL DD card keywords that are used when allocating an IAM dataset through JCL.

<table>
<thead>
<tr>
<th>KEYWORD</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATACLAS=</td>
<td>Specifies the name of the DFSMS Data Class being requested for this dataset. The file attributes will be obtained from this class. For DFSMS managed datasets, if the name includes the literal $IAM, the dataset will be an IAM dataset.</td>
</tr>
<tr>
<td>DSN=</td>
<td>Specifies the 1 to 44 character dataset name. For IAM datasets that are not DFSMS managed, the literal $IAM must be part of the dataset name. Temporary datasets begin with a single or double &amp;.</td>
</tr>
<tr>
<td>DISP=</td>
<td>Specifies the disposition of the dataset. All VSAM and IAM datasets, except temporary datasets, must be cataloged, and will be cataloged during step initiation when the dataset is allocated. Valid values include:</td>
</tr>
<tr>
<td></td>
<td>(NEW,CATLG) for permanent datasets, or (NEW,PASS) for temporary datasets.</td>
</tr>
<tr>
<td>EATTR=</td>
<td>Specifies whether or not this dataset can reside on an EAV volume. Values include:</td>
</tr>
<tr>
<td></td>
<td>NO – Cannot reside on EAV</td>
</tr>
<tr>
<td></td>
<td>OPT – Can reside on an EAV</td>
</tr>
<tr>
<td>KEYLEN=</td>
<td>For KSDS type of files, specifies the length of the key. This will override any value determined from a DATACLAS or MODEL if specified. For IAM, valid values are 1 to 249.</td>
</tr>
<tr>
<td>KEYOFF=</td>
<td>For KSDS type of files, specifies the offset of the key within the record (RKP). For IAM files, this must be less than or equal to 4091.</td>
</tr>
<tr>
<td>LIKE</td>
<td>Specifies the name of an existing IAM or VSAM dataset from which the RECORG, KEYLEN, KEYOFF, LRECL, and space attributes will be obtained.</td>
</tr>
<tr>
<td>LRECL=</td>
<td>Specifies the maximum record length. For KSDS type of files, this must be at least the value of KEYLEN + KEYOFF.</td>
</tr>
<tr>
<td>MGMTCLAS=</td>
<td>Specifies the DFSMS Management Class that is being requested for this dataset.</td>
</tr>
<tr>
<td>RECORG=</td>
<td>Indicates the type of dataset to define. Valid values for IAM are KS for a KSDS, ES for an ESDS or RR for an RRDS. IAM does not support the LS type of VSAM datasets.</td>
</tr>
<tr>
<td>STORCLAS=</td>
<td>Specifies the DFSMS Storage Class that is being requested for the dataset.</td>
</tr>
<tr>
<td>UNIT=</td>
<td>Specifies the type of device to which the dataset is being allocated.</td>
</tr>
<tr>
<td>VOL=SER=</td>
<td>Specifies the volume(s) to which the dataset is being allocated. Note that temporary datasets are limited to one volume. This is not required for DFSMS managed datasets.</td>
</tr>
</tbody>
</table>
The following are some examples of how to define IAM files through JCL. For clarity in
the examples, the program being executed is IEFBR14. However, any program could be
executed, including programs that may load and access the dataset.

Example H: Basic JCL Define of an IAM file

The example below is a basic example of defining an IAM KSDS type of file through
JCL. The dataset eligibility for an SMS managed volume will be determined by the
installations ACS routines.

```
//JCLDEFIN EXEC PGM=IEFBR14
//NEWIAMFL DD   DSN=new.my$IAM.file,DISP=(NEW,CATLG),
//   RECORG=KS,KEYLEN=8,KEYOFF=0,LRECL=128,
//   UNIT=SYSDA,VOL=SER=MYVOL1,
//   SPACE=(CYL,(20,2))
```

Figure 10-10: Basic JCL Define of an IAM File (EX1020H)

Example I: Define of an IAM Dataset with SMS Classes

In the following example, the file attributes are picked up from the DATACLAS, as well
as the request to make the file an IAM file.

```
//SMSDEFIN EXEC PGM=IEFBR14
//NEWIAMFL DD    DSN=new.myfile.cluster,DISP=(,CATLG),
//   DATACLAS=(MY$IAMF1),STORCLAS=(PERMVSAM)
```

Figure 10-11: JCL Define of an SMS Managed IAM Dataset (EX1020I)

Example J: Define of a Temporary IAM Dataset

The example below allocates a temporary IAM dataset, whose attributes are being
taken from an existing IAM or VSAM dataset.

```
//TEMPIAM  EXEC PGM=IEFBR14
//TEMPFILE DD   DSN=&amp;TEMP.$IAMFIL,DISP=(,PASS),
//   LIKE=my.indexed.cluster,
//   UNIT=SYSDA
```

Figure 10-12: JCL Define of Temporary IAM Dataset (EX1020J)
In the following example, an IAM ESDS type of file is being defined.

```plaintext
//ESDSDEF1 EXEC PGM=IEFBR14
//ESDSFILE DD   DSN=my.esds$IAM.cluster,DISP=(,CATLG),
//   RECORG=ES,LRECL=1020,
//   UNIT=SYSDA,VOL=SER=MYVOL1,SPACE=(CYL,(2,1))
```

*Figure 10-13: JCL Example Define of an IAM ESDS File (EX1020K)*
Defining IAM Datasets under TSO

IAM datasets can be defined under TSO, using a variety of methods. TSO offers an IDCAMS Define process through the DEFINE CLUSTER command. TSO also has a method similar to the JCL define, by using the TSO ALLOC command. IAM provides a set of ISPF panels, and the underlying software, which includes the capability to define an IAM dataset through an easy to use fill in the blanks method. Many other products offering various capabilities under ISPF for VSAM files will also work with IAM files.

The one major difference under TSO is that all of the above methods utilize the standard TSO dataset naming conventions. If the dataset name is not specified within apostrophes, then the dataset name will be prefixed. This prefix is normally your TSO user id, but can be changed.

One of the easiest ways to allocate an IAM dataset is through the IAM ISPF panels. They feature a fill in the blanks mechanism, along with providing easy specification of various IAM overrides all on one screen. To get to the IAM Define panel, select option I on the IAM Primary Option Menu. Be sure to fill in the dataset name field and the dataset type field. Optionally, you can specify an existing IAM or VSAM dataset as a model for the dataset attributes. Below is an example of the IAM Primary Option Menu with the user provided text highlighted.

---

![IAM ISPF Primary Option Menu](image)

**Figure 10-14: IAM Primary Options ISPF Panel**

---
Once you hit enter, the IAM Definition panel is displayed. If you had specified a MODEL dataset, the information from the model is filled in on the definition panel, any of which can be changed. This panel includes the various parameters and attributes that can be specified for the file. Note that this panel displays the fully qualified dataset name at the top left. The attributes essentially match those that must be specified on an IDCAMS Define Cluster request. Notice the More: + on the top right of the display. This indicates that additional fields are available, and that you scroll down to see them. (The panel as shown is fully inclusive.) Further down includes an area to specify the various IAM Overrides that affect the file definition. Once the necessary attributes are specified, then hit enter to allocate the file. Note that the data entered on the panel is highlighted text in the example.

Figure 10-15: IAM Dataset Define Panel (1 of 3)
Figure 10-16: IAM Dataset Define Panel (2 of 3)

Figure 10-17: IAM Dataset Define Panel (3 of 3)
If all the required parameters have been specified, and the allocation is successful, then the IAM Primary Option Menu will be re-displayed, with the status message in the top right corner. Notice the highlighted feedback, which indicates that the IAM dataset was successfully allocated. To verify that the file has been defined how you want it; press enter to have the IAM dataset attribute ISPF panel displayed.

![IAM Dataset Define Confirmation](image)
IAM responds with the IAM File Characteristics display, with the various attributes filled in. The format of the display is based on the format of the file being displayed. There are displays for IAM Enhanced format files, as seen below, a display for IAM Compatible format files, and a display for VSAM files.

![Figure 10-19: IAM ISPF Dataset Attributes Display (1 of 2)](image-url)
Figure 10-20: IAM ISPF Dataset Attributes Display (2 of 2)
IAM datasets can also be defined with the TSO Define Cluster command. While this does not provide the same ease of use as the IAM ISPF panels, it is an option if the IAM ISPF panels have not been installed. This command essentially is identical to the IDCAMS Define Cluster command, with the same format and parameters. Some of the abbreviations may be slightly different. Refer to the IDCAMS Define section for information relating to the parameters required for IAM datasets. Also, you can issue the TSO command: “HELP DEFCL” to get the information about the TSO DEFINE CLUSTER command.

Remember that the dataset name fields use the TSO naming conventions. The names will be prefixed unless enclosed within apostrophes.
After the file has been defined, you can use the TSO LISTCAT ALL command to see the file attributes. The following example shows how to issue the LISTCAT and the results.

![Figure 10-23: Example TSO LISTCAT Command](image)

**TSO LISTCAT ALL Output** As a result of the LISTCAT ALL, IAM produces what is called an IAMPRINT Report, which is written directly to the TSO terminal, which is followed by the standard IDCAMS SYSPRINT output. Refer to the section on IAM Reports for complete details on the IAMPRINT Report. Note that depending on your terminal, the output may require multiple screens, and spacing may appear slightly different.

![Figure 10-24: Figure 21: TSO LISTCAT Output (1 of 2)](image)
IAM files can also be defined with the TSO ALLOC Command. This is similar to defining IAM files through JCL. This command provides far fewer specifications than the DEFINE Command, allowing only the basic essentials to be specified. The ALLOC Command has the same parameters as added to the JCL DD card: RECORG, LRECL, KEYLEN, and KEYOFF. The use of a DATACLAS will add a few more fields, such as share options and free space. The highlighted text indicates the command and the operands. For complete information on the ALLOC command, use HELP ALLOC.

The ALLOC Command will either respond with error messages, or return with no messages at all, which indicates success. As with the prior DEFINE CLUSTER Command, you can issue a LISTCAT ALL just to validate the allocation.

As you can see, there are several methods available for defining IAM files. The most common method is to use the IDCAMS DEFINE CLUSTER command, however there are other choices. The other methods include through JCL DD cards, or through various mechanisms under TSO. The key ingredient is that you have to indicate that the file is to be an IAM file instead of a VSAM file. This is done by using OWNER($IAM), or by placing $IAM within the dataset name, or by using a DATACLAS or STORCLAS with $IAM in the class name. For many files, this is the only change required to use IAM instead of VSAM.
10.25 Using Hardware Compression

Overview
IAM supports a hardware data compression option to use the IBM Hardware Compression instruction. The IAM use of hardware compression is specified when an IAM file is defined or loaded by using an IAM CREATE Override of DATACOMP=HARDWARE (or HW). IAM offers a dynamic dictionary build function, which will be automatically invoked when a file is loaded with hardware compression requested, unless a specific customized dictionary is requested.

The dynamic dictionary build function will temporarily store the initially loaded records in a Data Space. Once sufficient data is accumulated, IAM will build a hardware compression dictionary then physically write the previously saved records out to the file in a compressed format. All subsequent writes to the file will have their records compressed as IAM processes the write requests. During the file close process, IAM will write out this dictionary into the IAM dataset itself for future processing with the dataset. Customers can also create their own compression dictionaries for IAM to use, and specify the dictionary name by the DICT= IAM Override. The original IAM provided dictionary is still available, and is specified by an IAM Create Override of DICT=4IAM.

IAM Software Compression
The IAM software compression technique uses a highly optimized proprietary software algorithm to compress data by eliminating strings containing repetitive byte values. Additionally, the IAM technique has very low cost for strings of data that could not be compressed. This technique, along with IAM being able to make as full use as possible of the capacity of each track, provided significant space savings for many datasets. Also, because this technique requires no dictionary, it reduces virtual storage and DASD storage requirements. While other compression techniques could provide greater compression, such techniques generally also came with a high price tag in terms of CPU consumption.

Hardware Compression
The IBM Hardware Compression algorithm relies on compression and corresponding decompression dictionaries. These dictionaries allow for compression and expansion of repeating data patterns within the records. A couple of difficulties are encountered with such an algorithm. First, for optimal compression, the data must be previewed to find the repeating data patterns within the dataset. Then from those repeating data patterns, the most frequently observed data patterns are converted into compression and decompression dictionaries. Also, one must insure that the dictionaries are safely stored, because if the decompression dictionary is lost, then the data cannot be decompressed thereby becoming useless. An alternative method of building or selecting a dictionary is to scan the first few records as they are being loaded, and select a generic dictionary that appears to offer best compression, or build one based on the initial data. Such dictionaries can provide decent amount of compression, although not the most compression possible for any given file.

IAM Use of Hardware Compression
IAM provides support for hardware compression, which can be selected as the default compression technique via the IAM Global Options table. The IAM software compression algorithm is still fully supported, and remains the default compression technique as the IAM Global Options are shipped. The hardware compression option can be selected either as the default compression technique or on an individual file basis by specification on an IAM CREATE Override. When IAM uses hardware compression, IAM will either dynamically build a dictionary based upon the data contents of the initially loaded records, or use a compression dictionary selected by the user via the DICT= IAM CREATE Override. In either case, IAM will store the dictionary within the dataset itself when the dataset is loaded.
The objective of the IAM dynamic dictionary build function is to provide users with an easy method to utilize the hardware compression functions to achieve beneficial data compression with minimal overhead in creating compression dictionaries. Users that require the best possible compression will in most circumstances achieve that only by the use of a customized hardware dictionary, which can be built as per the instructions provided below.

Creating a Compression Dictionary

There are a few steps required to build compression dictionaries for your files. The procedure consists of the following steps:

1. Create the control statements required for executing the IBM REXX EXEC that will read the data, and generate the dictionary. Review the information in `SYS1.SAMPLIB(CSRBDICT)`, which includes detailed instructions on using the exec.

2. Create a sequential dataset containing the data you want to build a dictionary for, or a representative subset of the data.

3. Execute the CSRBDICT REXX EXEC, with the sequential dataset you created previously, and with the control information you’ve decided on.

4. Assemble and link the dictionary generated by the CSRBDICT REXX EXEC into load module format.

Using CSRBDICT

Using the CSRBDICT REXX exec can be a rather intimidating task, as there are many parameters that can be specified. Determining the best settings for many of the parameters may require multiple executions of CSRBDICT, varying the various parameters, and then reviewing the results. The exec can run for a long time, using lots of CPU time to come up with a dictionary. It is easiest to start with a very basic execution and testing the resulting dictionary. If you are satisfied with the amount of compression you obtain, then go with that. For many files, this basic approach can yield excellent results. If you are looking for more compression, or attempting to create a dictionary for multiple datasets, then you can get more involved with varying parameter settings, and providing a more detailed layout of your data records to CSRBDICT.
To get you started, a basic set of control card input is being provided in the IAM ICL (Installation Control Library). The member name is BDICTEX1, and is shown below. This example uses the basic IBM recommended parameters, and provides two field statements. The first field statement describes the data up to and including the key, which is going to be ignored by CSRBDICT because IAM will not attempt to compress that data. The second field statement is for the rest of the data in the record, which will actually be subject to compression. All you need to do is to alter the starting position on the second field card to indicate the appropriate position of the first byte after the key within the data file for which the dictionary is being built. When selecting the values to use in the CSRBDICT "spec" file, for the "dicts" field you must specify either “AF ASM” or “AFD ASM” so it will generate a file containing assembler language representation of the dictionaries.

**The following is an example for building a 4k entry dictionary**
**just using a basic pattern scan. The first field card indicates to skip**
**the data that is up to and including the key. The second field card is**
**for the rest of the data in the record.**

```
**results maxnodes maxlevels msglevel stepping prperiod dicts**
  r        40000   64   3   f 7 2 7 1000      afd asm
**colaps opt treedisp treehex treenode dupccs**
  aam      opt     x   h   n      x
**FLD col type dcmem     INT     intspec**
  FLD 1    ns
  FLD 15   sa
  FLD end
```

Figure 10-26: Example input to CSRBDICT REXX exec

The next step is to obtain a representative sample of the data that is contained within the file that you are going to be creating the dictionary for. If the file size is relatively small, say under 50,000 records, you can probably use the entire file. However, if it is larger or if CSRBDICT is taking too long to run, you can either take a sample of the records in the file, or revise the “stepping” value in the BDICTEX1 member so that the entire input file will not be used. So, for example to reduce the amount of data scanned to say a little less than half the data (in this case 3/7), change the stepping values from “f 7 2 7” to “f 3 2 7”.

For execution parameters, you must also specify format-1 sibling descriptors, which is done by specifying the value 1 for the “sdfmt” field. For dictionary size try starting with 4(K) entries. Depending on your data patterns, you may find that a larger or smaller dictionary size will yield better results. The CSRBDICT process can take a long time to run, so be patient. Shown below is an example of the command to execute the CSRBDICT REXX exec.

```
ex 'sys1.samplib(csrbdict)’ ’4 1 eb “my.test.data” (“IAM.ICL(BDICTEX1)”’
```

Figure 10-27: Example of executing CSRBDCIT REXX exec
The CSRBDICT REXX exec can also be executed from a batch job. Shown below is an example of JCL to do just that:

```
//TSOTMP EXEC PGM=IKJEFT01,DYNAMNBR=60,TIME=120
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
  PREFIX myuid
  EX 'SYS1.SAMPLIB(CSRBDICT) '4 1 EB TEST.DATA (IAM.ICL(BDICTEX1))'
```

After successfully running CSRBDICT, there will be several output files. There are two files of primary interest to the IAM dictionary build process, and they will have the suffixes of ACDICTs1 and AEDICTs1, where "s" will be the number of K entries in the dictionary, either 1, 2, 4, 8 or H for 512 entries. These two datasets will be assembled and linked into a load module that can be used by IAM for the compression and expansion (decompression) dictionary.

In the IAM ICL (Installation Control Library) is an example, HWDASM, of the JCL to assemble and link the dictionaries for use by IAM. The first step ASMACD assembles the compression dictionary. Change the name of the SYSIN dataset to the name that was created by CSRBDICT for your file, it will have the suffix ACDICTx1. The second step, ASMAED, assembles the expansion dictionary. Change the SYSIN dataset to the name that was created by CSRBDICT for the expansion dictionary for your file. It will have a suffix of AEDICTx1. The third and final step, LKED, will link the two dictionaries together into a load module that IAM can use. The first four characters of the dictionary name must be ‘IAMD’, and you can choose the last four characters. Make sure that the characters you use do not conflict with any existing load module names or other dictionaries. A recommendation is to set the first optional character to the “s” value of H, 1, 2, 4, or 8, and then select three other alphanumeric characters. It is recommended that you place it in a load module library other than the IAM library, so that way it will not be lost when a new level or version of IAM is installed.

To use the dictionary that you just created, define an IAM file using an IAM Override card specifying hardware compression, and a dictionary name with the last four characters that you selected for the dictionary name. For example, if you chose 4ABC in the prior step, use the following override:

```
//IAMOVRID DD *
  CREATE DD=&ALLDD,DATACOMP=HW,DICT=4ABC
/*
```
You can now load your file using your compression dictionary. After the file is loaded, if you perform a LISTCAT, the IAMPRINT report should indicate that the dataset is hardware compressed, with a dictionary name of 4ABC, and that the dictionary is stored in the IAM dataset.

After seeing the amount of space used, you may want to try changing some of the parameters for CSRBDICT to see if you can obtain better compression. If so, change the parameters and rerun the build process, making sure that you’ve noted your prior parameters and the results. Once you are happy with the results, you can save your final resulting dictionary.
Loading IAM Datasets

After successfully defining an IAM Dataset, the next step is to load data into the dataset. Generally, the file load process is where the dataset is populated with records, and the index structure is built. For IAM datasets, this portion of the index is called the prime index. Once the load is completed, the prime index for the dataset is established, and will not change. The load process is restrictive in that the only I/O operation allowed is to write new records into the dataset. For KSDS type of files, the records must be written in ascending key sequence. The file load must write one or more records to the dataset, and subsequently successfully close the dataset for it to be considered a loaded dataset. Once loaded, the dataset can be processed with the full range of I/O operations.

IAM datasets can be loaded by any program designed to load VSAM clusters. This includes system utility programs, such as IDCAMS REPRO and the SORT, as well as application programs. Each I/O request must be checked for successful completion, including the OPEN and the CLOSE. As will be explained in the subsequent paragraphs, critical file structure related processing occurs during Open and Close processing. If these processes do not complete successfully the dataset cannot be used. Failures during the file load process will generally result in the dataset having to be reloaded.

Open Processing

When the IAM dataset is opened for the file load, IAM reads in the attributes, parameters and overrides that were specified when the file was defined, and checks for current overrides. Any IAM overrides provided for the file load will supersede the values specified when the file was defined. If there were any errors detected by the override processing, the file load will fail. This is done to prevent a dataset from being improperly loaded with incorrect attributes, which can result in subsequent problems. Using that merged information set, all of the file attributes are validated and established. Conflicts and errors within the merged attribute set will cause the OPEN to fail with a return code of 8, and an appropriate error code. Because most of the errors will be caught during the define process, file attribute errors are very rare during the open process. If one does occur, there will be an error message in the Job Log indicating the reason for the error. After the Open process successfully completes, the program can begin writing records into the dataset.

Record Processing

As each record is written, IAM validates the length, and for KSDS type files does key sequence checking. A key sequence error is not considered by IAM to be a fatal error for a file load, although it may indicate a fatal error for the application. For ESDS type of files, IAM calculates an RBA value to be used as the internal key. The RBA value is based on the uncompressed size of the record, the RBA of the prior record, and the requested CI size. If the IAM data compression feature is enabled for this file, IAM will compress the record. As long as the compressed length is less than the uncompressed length, the compressed form of the record is written to the file. As each block is filled, leaving room for the Integrated Overflow (CI% free space), the block is scheduled to be written, and the highest key in each block is temporarily saved in either a data space, or in a dynamically allocated work file. Additional DASD space for the IAM dataset is acquired as necessary.
After all the data has been written to the file, the application program must explicitly close the dataset. When the Close is issued, IAM will then read through the temporarily saved high key structure, and format the index section of the file. If there is a hardware compression dictionary, it is written out to the IAM dataset. When this process is successfully completed IAM returns to the calling program, and the dataset is now ready for use. Because errors can occur during this index build process, in particular insufficient DASD space to hold the index, it is important to check for a successful completion of the Close. An error during Close will result in a return code of 4, and an appropriate error code returned in the ACB.

A unique processing capability of fixed length record RRDS (numbered) files is that they can be loaded in a random relative record number sequence. The records are actually stored in ascending relative record number sequence. VSAM will pre-format files when a higher relative record number is written, and then will go back and reread and write out control intervals as lower relative record numbers are written. This can result in very high EXCP's when loading these files with VSAM. IAM takes a different approach. When a fixed length record RRDS is being loaded randomly, IAM will temporarily store the records in a data space. If necessary, multiple data spaces will be used. When the file load process closes the data set, IAM will then write the records out in ascending relative record number sequence. Because of this function, customer's planning on using fixed length RRDS files that are loaded randomly will need to make sure that their installation exits provide for adequate data space usage for IAM to load these data sets.

Another consideration for fixed length record RRDS files is that IDCAMS will ignore any values specified for FREESPACE on the define request. This is because VSAM reserves a fixed length slot within the cluster for each record. IAM does not reserve space for unused relative record numbers. To compensate for this, customers can either use the RRDSCI% Global Option to indicate a default amount of free space per block, or can use the I= (Integrated Overflow) IAM Override when defining or loading a fixed length record RRDS file to reserve some free space to allow for inserted records, or updated records where the length may change due to data compression.

IAM also supports loading variable length RRDS files in random relative record number sequence. An IAM override must be specified to do this because normal VSAM does not support this capability. To use this feature, specify on an IAM Override RRDSLOAD=RANDOM. As with fixed length record RRDS files, IAM will use data space(s) to hold the records until the file is closed for load processing.

IAM supports data compression for RRDS data sets, using either the proprietary IAM software compression, or using the IBM hardware compression instruction. This may provide for significant space savings for IAM RRDS files when compared to VSAM RRDS files.
Buffering

The file load process is essentially a sequential output process. To optimize the I/O process, IAM uses a different buffering technique than the Real Time Tuning that is used on normal file access and update. During open processing, IAM acquires a pool of buffers to be used, based on the default or overridden value for CRBUFOPT. The buffer pool is split into two sets. When the first set of buffers have been filled with data, IAM then issues the I/O for those buffers. Processing continues with the application passing records, which are placed into the second set of buffers. IAM is effectively providing I/O and processing overlap. When the second set of buffers is filled, IAM issues the I/O request to write all of those buffers, and waits if necessary for the I/O on the prior buffer set to complete. When that I/O completes, IAM continues accepting records into the first set of buffers. For efficient physical I/O, IAM always writes out data in full track increments. The largest number of buffers that will be used for a file load is specified by CRBUFOPT=MCYL. In that case, IAM acquires enough storage for two cylinders worth of data, and will write out a complete cylinder per physical I/O. The fastest file load is obtained by using CRBUFOPT=MCYL, which is the default setting from the IAM Global Options. Remember to supply buffers for the input file as well.

64_Bit Virtual Storage

For circumstances where there is some virtual storage constraints, users can request the IAM load process acquire the buffers in 64-bit addressable virtual storage. This is done by specifying an IAM Create Override of CRUBOPT=64BIT. This will result in IAM acquiring 2 megabytes of 64-bit virtual storage for the buffers enough to hold 2 cylinders of data. The buffering technique is identical to the CRBUFOPT=MCYL, only the location of the buffers is different.

z/HPF High Performance FICON

IAM will automatically use z/HPF High Performance Ficon channel programs when zHPF is enabled on the system and the device that the data set is being written to supports the level of zHPF required for EXCP processing.

File Full, or Sx37 Error

IAM file load processing is protected by other products that will handle x37 type of abends. Such products change the secondary space allocations and can add additional volumes to the dataset to avoid the x37 abend. If the recovery is not successful the x37 abend will still occur.

A file full error caused by an x37 abend during the load process is considered a fatal error by IAM. Because the index is written at the end of the dataset, an out of space condition will prevent the index from being properly written out to DASD. There will usually be messages indicating some type of Sx37 error condition. To insure that IAM is able to properly clean up after such a failure, in particular to release the ENQ that are issued, IAM will attempt to avoid an actual Sx37 abend under CICS. In that case, the request will fail with a file full logical error. A file full error will also be raised if the area being used for the temporary storage of the high keys is filled. This is considered a fatal error to the file load. Either increase the storage available for the data space by using the DATASPACE override or if not using the data space increase the allocation values for the work file in the IAM Global Options Table, with the WORKPRIMARY and WORKSECONDARY values.
Special Considerations

Some applications use a technique for loading files that consists of loading a single record, followed by a mass insert. In some instances this technique is obvious due to an IDCAMS REPRO of a special record into the file. Other times it is hidden. In particular, this process occurs when a COBOL program does an OPEN OUTPUT with ACCESS IS DYNAMIC or ACCESS IS RANDOM specified. What COBOL does in this circumstance is load a record with a key of binary zeros and then closes the file. COBOL subsequently reopens the file and deletes the record. This effectively places the file in a loaded but empty state.

The result is that all the records that are mass inserted into such file are contained within the overflow areas, Extended Overflow and Extended PE. The storage required for the index to these overflow areas can be substantially more than if the file had been fully loaded rather than single record loaded. If the single record load process cannot be changed, then it is recommended that the dataset be reorganized after the mass inserts are done. This will reduce overall virtual storage requirements, and provide for more efficient processing.

JCL

In general there are no required JCL changes from file loads that were using VSAM clusters. The DD card for the IAM dataset should specify only the DSN= (Dataset name) and a DISP=OLD (disposition). A DISP=SHR is allowed as well, although DISP=OLD is recommended because only the program loading the dataset is allowed to have the dataset open while the load is in progress. IAM will enforce this restriction using an ENQ mechanism with the major name of IAMENQ. The AMP parameters may be specified however there definitely must not be any DCB parameters. The VOL=SER and UNIT parameters should not be specified either.

When the IAM dataset is defined and loaded in the same IDCAMS step, the REPRO should use the OUTDATASET (ODS) parameter, not the OUTFILE parameter to reference the IAM dataset. This is because if the IAM dataset was defined to a different volume than the one it existed on when the job step was started, then the DD card is no longer valid because it has the old volume(s) allocated to it. This guideline has to particularly be followed by installations that are using DFSMS managed datasets, or some other allocation products such as BMC’s MAINVIEW (formerly POOLDASD), CA-ALLOCATE, or ACC/SRS.

The IAMINFO DD is optional, but highly recommended. A report will be produced to that DD when the IAM file is closed, containing the dataset attributes and statistics about the load. The IAMOVRID DD statement is only necessary when providing IAM overrides for this job step. The IAM Overrides that are applicable to the file load are described below.

Long Key Compression

With this feature, IAM provides index compression for files with key lengths that exceed 128 bytes. For a 33-gigabyte file from a customer that had a key length in excess of 128 bytes, the uncompressed index required 227 megabytes. That was reduced by the IAM Long Key Compression to 49 megabytes, a 78% reduction! The IAM Long Key Compression may also provide storage savings due to improved compression with keys of shorter lengths, such as 64 bytes or 32 bytes because more keys can be grouped in a compression set than before.
As IAM is shipped, all files with keys of 33 bytes or longer are automatically eligible for long key compression. This can be changed with the IAM GLOBAL OPTION of LONGKEYS=. This Global Option specifies that keys with lengths equal to or greater than that value will be automatically selected for Long Key Compression. Valid values are from 8 to 255. The maximum recommended value is 128, which was the default in prior releases.

The IAM CREATE Override of INDEXCOMPRESS now has a new value of LONG. In addition to the prior YES or NO values, a value of LONG can be specified to request that IAM create a long key compressed index for that particular file.

Overrides for Dataset Load

There should only be an infrequent need to override the file load process. The main reasons for using an override for the load process are to use the IAM feature of reorganizing the file without decompressing the data, to alter the buffering for the file load, to increase the size of the data space used to temporarily hold the index, or to specify certain attributes if the dataset had been defined through JCL. The BACKUPCOMRESSED, CRBUFOPT, DATASPACE and RRDSLOAD keywords are only applicable to the file load process, and when needed must be specified for the load.

The following list contains the CREATE Override keywords which are applicable to the file load process. For the most part, the CREATE Override keywords have the same meaning and implication when used during the file define. The underscored portion of the keywords indicates their minimum abbreviation. The keywords applicable to the define process are:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BACKUPCOMRESSED</td>
<td>Specifies that the input data is already in an IAM Software Data Compressed format. This can only be specified on a file load step with data that was created with the IAM ACCESS Override of BACKUPCOMRESSED. The Backup Compress feature is only valid for files compressed with IAM software compression.</td>
</tr>
<tr>
<td>BLKSIZE=nn</td>
<td>Specifies the block factor (1 - 15) or block size of the IAM dataset.</td>
</tr>
<tr>
<td>CRBUFOPT=</td>
<td>Specifies the buffering option to be used for the file load.</td>
</tr>
<tr>
<td>CYL</td>
<td>Buffers for one cylinder's worth of blocks is acquired, approximately 1/2 cylinder is written per I/O.</td>
</tr>
<tr>
<td>MCYL</td>
<td>Buffers for two cylinder's worth of blocks is acquired, one cylinder is written per I/O.</td>
</tr>
<tr>
<td>MTRK</td>
<td>Buffers for ten tracks are acquired, five are written per I/O.</td>
</tr>
<tr>
<td>TRK</td>
<td>Buffers for two tracks are acquired, one track is written per I/O.</td>
</tr>
<tr>
<td>64BIT</td>
<td>Similar to MCYL except the buffers are in 64-bit virtual storage.</td>
</tr>
</tbody>
</table>

Default value is MCYL
Keyword | Brief Description
---|---
**DATACOMPRESS**= | Indicates whether IAM is to data compress this dataset. Valid values are:

**YES** – Indicates that the dataset will have its data compressed, using the default data compression technique from the IAM Global Options Table.

**NO** – Indicates that no data compression will be used for this dataset.

**HW** – Indicates IAM will use the z/OS hardware compression instruction to compress the data. If the DICT= override specifying a compression dictionary name is not specified, IAM will dynamically build a compression dictionary when the file is loaded, based on the data in the first several records.

**SW** – Indicates that IAM will use its standard software compression algorithm. No dictionaries are used with this form of compression.

The default is based on the IAM Global Options Table, which as shipped is any dataset that is 75 tracks (5 cylinders) or larger is eligible for IAM data compression. The default technique is also based on the IAM Global Options Table, which is shipped as software compression.

**DATASPACE**= | Specifies the size, in megabytes, of the Data Space to be used for the temporary storage of the index to the IAM file that is being loaded. Valid values are from 0 to 2048. A value of 0 results in the use of a dynamically allocated temporary dataset. The default value is 2048 based on the IAM Global Options table.

**DDNAME**= | Specifies which dataset the override is applied to. This will indicate the name of the DD statement being opened for the file load. If this value is set to **&ALLDD**, then the overrides are applicable to any IAM file that is not otherwise explicitly overridden.

**DICTIONARY**= | Specifies the four character suffix for the name of the user provided hardware compression dictionary. The dictionary must be in load module format, with the first four characters being ‘IAMD’. Review the section on Hardware Compression for information on creating and naming the compression dictionary. The default is to use the IAM provided dictionary.

**DSN**= | As an alternative to **DDNAME**=, specifies the dataset name that the override data is to be applied to. Either DSN= or DDNAME= must be specified.

**ENHANCED** | Specifies that IAM is to create an Enhanced format IAM dataset. This type of dataset uses a dynamic overflow area, which can acquire additional DASD extents during file updates, as needed. This is the default value, unless otherwise specified or changed in the IAM Global Options Table.

**INTEGRATED=nn** | Specifies CI% free space, especially useful for data compressed ESDS datasets.
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Brief Description</th>
</tr>
</thead>
</table>
| INDEXCOMPRESS=      | Specifies whether or not the file is eligible for a compressed index. Valid values are YES, LONG or NO.  
  
  When YES is specified, the file still has to meet the minimum size requirement as indicated by Global Options or CORELIMIT override, the key length must be between 4 and 128, and the index compression must yield storage savings of at least 10%.  
  
  When LONG is specified, IAM will use a Long Key Compression algorithm, which does not limit a compression set size to 256 bytes. This will provide index compression for files with keys longer than 128 bytes, and may also provide other files with keys of 32 bytes or longer with better compression  
  
  When NO is specified, a compressed index will not be created for the file. Default value is YES except for files with keys of 33 bytes or more which will default to LONG. |
| JRNAD=              | Specifies the IAM journaling capabilities to be used.  
  
  For files not being processed by IAM/RLS, IAM/PLEX or not using IAM/RLS or IAM/PLEX journaling, users must allocate and catalog a log dataset to be used for the journaling, which is required to be the name of the IAM dataset / cluster, appended with the characters ".LOG". IAM journaling will not be active during file loads, reorganizations, or during recovery from the journal.  
  
  For files being processed or journaled by IAM/RLS or IAM/PLEX, the users must specify the journal datasets to the IAM/RLS or IAM/PLEX startup procedure.  
  
  Valid values are:  
  
  BOTH or ALL- The IAM log dataset will contain both before and after images. This will enable the user to perform either a forward recovery, or a backward (backout) type of recovery.  
  
  BEFORE - The IAM log dataset will contain before images of updated records. This option allows backward (backout) recoveries only.  
  
  AFTER - The IAM log dataset will contain after images of updated records. This option allows forward recoveries only.  
  
  NONE - The IAM journaling feature will not be used for this IAM dataset.  
  
  Default value is NONE. |
| MAXBUFNO=nnnn       | Specifies the maximum number of buffers to be used during file access for this dataset. |
| MAXSECONDARY=nn     | Specifies a value from 0 - 10 as a multiplication factor for the secondary space quantity, when the dataset exceeds five extents on a volume. When overridden on the load, this will be the value for extents taken only during the file load. |
Keyword | Brief Description
--- | ---
**MINBUFNO=nnnn** | Specifies the default minimum number of buffers to be used during file access for this dataset.

**MULTIVOLUME=**  
[ PRIMARY | SECONDARY] | Specifies space quantity to request when IAM believes that the next extent will be placed on the next candidate volume.

**PRO=[YES | NO]** | Specifies whether or not IAM will use the Prime Related Overflow (PRO) for this dataset. This format is only recommended for use on datasets that will have close to a million or more records in overflow and if shared, must be used under IAM/RLS or IAM/PLEX. Valid values are:

**YES** – Use the PRO format for this dataset.

**NO** – Do not use the PRO format for this dataset.

**PSEUDORBA** | For ESDS files, indicates that the file can exceed 4 gigabytes of user data. IAM generates 4-byte RBA values that are different than normal VSAM values. Cannot be used with software that depends on the normal VSAM values. Also see XESDS override described below.

**RELEASE=[YES|NO]** | Indicates whether or not IAM is to release unused and unreserved DASD space whenever the file is loaded. Default is to release space on the first load only, and only when a secondary space value is specified. COMPAKTOR will also release unused space for IAM enhanced format datasets until the file has expanded into the extended overflow areas of the file. If the Global Option Table has set the default DSORG=DA, space will not be released as of z/OS 1.11 due to IBM changes in this process.

**RLSID=[nnnn]** | When specified with no operand, indicates that this dataset is to be processed using IAMRLS, regardless of Share Options or the IAMRLS Dataset Name Tables. The operand, if specified, is the four character identifier of the preferred IAMRLS address space that this dataset is to be processed by. Note that this is for future accessing of the dataset, as a file load will never be performed under IAMRLS.

**RRDSLOAD=**  
[SEQUENTIAL | RANDOM ] | When loading a fixed length record RRDS in random mode, but in an actual ascending relative record number sequence, specify RRDSLOAD=SEQUENTIAL to prevent IAM from using data space(s) to temporarily hold the records.

When loading a variable length record RRDS in random mode, specify RRDSLOAD=RANDOM to allow that process, as VSAM does not support random loads of variable length record RRDS files.

**VAROVERFLOW=[YES|NO]** | For Enhanced format files, indicates that IAM can use true variable length records in the Extended Overflow area. This will provide for more effective use of DASD space. The default value is YES.

**XESDS | NOXESDS** | For ESDS files, indicate that IAM is to use an 8-byte RBA value (XESDS) or a 4-byte RBA (NOXESDS). The XESDS option is consistent with the VSAM Extended Addressability support, and allows ESDS files to exceed 4 gigabytes of data, but application programs must be able to handle the 8-byte RBA. Default is NOXESDS, IAM will use a 4-byte RBA based on the IAM Global Options.
Examples of Loading an IAM Dataset

Below are a couple of examples using IDCAMS to load an IAM dataset. As stated above, IAM datasets can be loaded by any program that can load a VSAM cluster. For conversions from VSAM clusters to IAM datasets, there rarely will be any need to change the JCL, except perhaps to add an IAMINFO DD for SYSOUT to get the detailed report of the file load.

In the example below, an IAM dataset is loaded with IDCAMS. An IAMINFO DD statement has been added to obtain the run time file load report from IAM. The OUTFILE parameter is used to specify the output IAM file. Note the REUSE is specified on the REPRO command, which will cause the dataset to always be reloaded, even if it already contains data.

```
//LOADFILE EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=* 
//IAMINFO DD  SYSOUT=* 
//INFILE  DD   DSN=my.seqfile,DISP=SHR 
//IAMFILE DD   DSN=my.iamfile,DISP=OLD 
//SYSIN    DD    * 
    REPRO INFILE(INFILE) OUTFILE(IAMFILE) REUSE 
/*
```

`Figure 10-29: Example of Loading an IAM Dataset (EX1030A)`

In this example, the OUTDATASET (ODS) is used instead of OUTFILE on the IDCAMS REPRO. This will cause IDCAMS to dynamically allocate the IAM dataset. Because the ddname is not known in advance, the CREATE override specifies the dataset name (DSN=). In this example, the sequential input dataset was previously created by copying an IAM software compressed dataset with the BACKUPCOMPRESSED override, so it must be reloaded with that same override specified on the CREATE override. A large BUFNO value is specified on the DCB for the input file to provide for faster processing.

```
//LOADCOMP EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=* 
//IAMINFO DD  SYSOUT=* 
//SEQFILE DD   DSN=my.compressed.seqfile,DISP=SHR, 
             DCB=BUFNO=60 
//IAMOVRID DD   * 
    CREATE   DSN=my.iamfile,BACKUPCOMPRESSED 
/*
//SYSIN    DD    * 
    REPRO INFILE(SEQFILE) ODS(my.iamfile) REUSE 
/*
```

`Figure 10-30: Example of Loading an IAM dataset with compressed data (EX1030B)`
Processing IAM Datasets

Overview

After the IAM dataset has been defined and subsequently loaded with data, the IAM dataset is ready to be used for typical online and batch processing. Processing an IAM dataset instead of a VSAM cluster will generally significantly reduce CPU times, physical I/O’s, and elapsed times for most programs. The IAM VSAM interface was developed with the intent of eliminating, as much as possible, the need to change any program, JCL, or CICS tables. On rare occasion, there may be a need for a change, which can frequently be accomplished with an IAM override. IAM supports almost all of the typical VSAM processing options, including specification of LSR pools, user exits specified via the VSAM EXLST macro, asynchronous I/O, and so forth.

IAM does not support RBA or control interval access of KSDS types of files, and does not support control interval updates for ESDS type files.

Open Processing

As part of the functioning of IAM’s VSAM Interface, IAM will screen all open requests for VSAM ACB’s. If the ACB being opened is for an IAM dataset, IAM routes control to the appropriate IAM module to perform the OPEN processing. If the open is not for an IAM dataset, the IAM routes control to the IBM VSAM open processing modules, and does not further screen or interfere with VSAM open or I/O processing.

One of IAM’s major features is that the index to the file is kept in virtual storage while the file is open. This feature not only eliminates I/O for the index component, but also provides for savings in CPU time. IAM can always get to any specifically requested record with no more than one I/O. Because the index is read into storage at open time, open processing for an IAM dataset may take longer than a VSAM open. An IAM open also involves acquiring virtual storage required for processing, acquiring buffers, and setting up the I/O control blocks, and loading the necessary IAM load modules. IAM’s self-defining file structure eliminates the need for retrieving information about the dataset from the system catalog, which eliminates that VSAM overhead.

Reading the index to the prime area and the prime extension is a straightforward process. Building the overflow index involves more processing, because the overflow index itself is not saved within the IAM dataset unless PRO is being used. The basic overflow index contains one entry for each record in overflow, unlike the prime and prime extension index that contains the high key in each block. To build the overflow index, IAM must read all of the used overflow blocks, determine the key of each overflow record, and insert the key into the proper position in the overflow index. As more and more records are put into overflow, this process will take longer to complete. Regular file reorganizations will help to keep the open processing time at a minimum.

IAM Open processing also automatically performs an implicit verify operation. On files that have been opened for update, IAM will update the extended index and end of file information if necessary.

Once the file open process has completed, IAM is ready to handle the application I/O requests.
I/O Request Processing

I/O to an IAM dataset is performed by using the standard VSAM application programming interface (API). Standard VSAM protocol is used by IAM, including error reporting. IAM does not abend when an I/O request fails, the appropriate return codes and failure codes are provided, and the appropriate error exit routine is also invoked if so specified by the calling program. As with VSAM, it is the responsibility of the application program to determine the appropriate action for any particular type of error. The various error codes set by IAM are documented in Section 80.21 of the IAM manual.

For KSDS types of files, IAM handles all requests except those that specify Relative Byte Addressing (RBA) or Control Interval access. Such requests are failed with an error code of 104 (x’68’) and a return code of 8. For ESDS files, IAM does not support Control Interval updates, which will fail with an error code of 68 (x’44’) and a return code of 8. For fixed length record RRDS files, IAM supports control interval reads and updates when the record size and control interval size are such that only one record will fit within a control interval.

Buffering

IAM’s dynamic buffer management technique, Real Time Tuning, works automatically in response to the application I/O requests. The IAM buffering technique dynamically adjusts buffering and I/O techniques, along with adjusting the number of buffers, based on the types of requests being issued by the application. IAM takes the guesswork out of the tuning process. There is no need to be concerned about whether or not to use LSR type of buffering, or how many buffers are needed for optimum I/O performance. IAM will select the buffering technique and quantity, within specified or defaulted limits, automatically. Buffering and I/O techniques are adjusted dynamically in response to the application requests.

Real Time Tuning works within a range for quantity of buffers, referred to as MINBUFNO and MAXBUFNO. These ranges can be specified through the IAM Override facility, using the MINBUFNO and MAXBUFNO keywords. Default values for MINBUFNO and MAXBUFNO are determined from the IAM Global Options Table. Using the default Global Option values will eliminate the need for increasing the MAXBUFNO value for all but the most heavily accessed datasets. The MAXBUFNO value can also be increased by the BUFND parameter specified in JCL as a sub-parameter of the AMP parameter, or within the program in the ACB.

64-bit Virtual Storage

The default as shipped is that IAM will use 31-bit addressable storage for the buffers it acquires. IAM can optionally use 64-bit addressable virtual storage when there are storage constraints. The use of 64-bit storage for buffers is specified by either the IAM ACCESS Override of BUF64=YES or by enabling the IAM Global Option BUF64. For users concerned about saving CPU time, please be aware that the use of 64-bit virtual storage for buffers does increase the CPU time due to z/OS page fix processing.

For more information on IAM’s Real Time Tuning capability, refer to Section 2.40 of the IAM Manual.

Turbo Mode Buffering

 Turbo mode is invoked during periods of heavy I/O activity on any particular IAM file, and will increase the frequency of acquiring additional buffers up to the MAXBUFNO limit, so that IAM will be more responsive to periods of heavy I/O requirements. With the IAM Global Options as shipped, Turbo mode will be invoked as necessary. Coupled with the default limits for MAXBUFNO Turbo Mode will help reduce the need for providing MINBUFNO and MAXBUFNO overrides.
z/HPF High Performance FICON

IAM will automatically utilize z/HPF I/O architecture channel programs when zHPF processing is enabled in the system, and when the operating system and DASD devices being used support the z/HPF functions required for use by EXCP. This I/O architecture provides for more efficient use of the FICON channels, increases channel capacity to support more I/O activity, decreases channel connect time, and may reduce elapsed times for jobs that can utilize this function.

Caching Overflow Blocks

For files with large overflow areas that are being sequentially processed, IAM has a feature that uses z/OS 64-bit virtual storage, referred to as above the bar storage. This capability can help improve run time for sequential processing, including backups and reorganizations of Enhanced Format IAM files with a large number of records in overflow. During OPEN processing, IAM reads all of the overflow area blocks to build the record level index. With this new caching feature enabled, the overflow blocks will be copied to an above the bar virtual storage area for later reference during file access. When one of these blocks is needed for a data record, IAM will copy the data block from the above-the-bar storage area into a normal I/O buffer, rather than performing a physical I/O.

The benefits for installations with sufficient real storage are reduction in physical I/Os and elapsed times to read the file sequentially. Installations without sufficient real storage will likely find that the time spent in paging will be detrimental to job performance. To use this new feature, simply specify the IAM ACCESS Override of CACHE64. IAM will calculate the amount of above-the-bar storage required to hold the overflow blocks, and acquire it. Many users may also need to specify the MEMLIMIT= parameter on their execute cards to make above the bar storage available to the job step, for example MEMLIMIT=4G for 4 gigabytes of above the bar storage.

Dynamic Tabling

Dynamic Tabling is a special feature of IAM that keeps the most frequently referenced randomly read records in a virtual storage cache. It is a different concept than buffering which handles blocks containing multiple records. Dynamic Tabling is record oriented, and is useful when a subset of the records scattered throughout the file are repeatedly referenced, potentially reducing the virtual storage needs to have a large number of buffers. This facility provides substantial benefit particularly for some online applications, by eliminating physical I/O as records are repeatedly retrieved from the table. Applications that may potentially benefit from this feature are those that have a high volume of GET RANDOM requests, combined with a very low quantity of PUT UPDATE and ERASE requests. The update requests are detrimental because they will force IAM to read the data block into storage, to perform the update. Other types of I/O requests will not be impacted by the use of the dynamic table.

Dynamic Data Space

The primary choice for placement of the dynamic table is the IAM Dynamic Data Space feature, where the dynamic table is placed in a z/OS Data Space. The use of a Data Space for the dynamic table allows for a larger number of records to be placed in the table. For the Dynamic Data Space, IAM has revised the algorithms used for storage management, record management, and the record search to provide for more effective utilization of storage resources, a faster record search capability, and reduced CPU overhead when inserting and removing records from the table. When records need to be removed from the table to make room for more recently referenced records, IAM now utilizes a true least recently used algorithm for selecting the record(s) to be removed.
The use of the Dynamic Data Space is on a file by file basis, with each individual data set for which the function is requested using its own data space. The capability is easily implemented by the use of the IAM ACCESS Override of DYNDS. Using the DYNDS= override, users simply specify the amount of storage that IAM is to use for the Dynamic Data Space in megabyte increments, up to a maximum of 2048, or 2 gigabytes. The virtual storage required for management of the Dynamic Data Space will come out of the Data Space itself so that use of the Dynamic Data Space will not impact virtual storage resources within the address space processing the dataset itself.

**DYNCORE**

DYNCORE is the original IAM dynamic tabling facility, which uses above the line virtual storage within the requesters address space. Therefore the amount of virtual storage is more limited than with the Dynamic Data Space. DYNCORE can be used on all IAM file formats, and is useful when the table size requirements are not too large.

This capability is easily used. Simply specify the table size, in kilobytes, using the IAM ACCESS DYNCORE override. Statistics on the DYNCORE usage are provided in the IAMINFO report.

**File Expansion**

IAM Enhanced Format files are dynamically expandable, through the use of secondary extents. IAM files that are not SMS Extended Format can be extended up to a maximum of 16 extents per volume for a maximum of 944 extents across 59 volumes. IAM files that are SMS Extended Format can be extended up to a maximum of 123 extents per volume, for up to 59 volumes. IAM Enhanced Format datasets do not have an explicit total maximum number of extents. As records are added, or updated records increase in length, additional DASD space is acquired and formatted as needed. This expansion area is called the IAM Extended Area. The Extended area consists of Extended Overflow blocks and Extended PE blocks. The Extended PE blocks are used to handle the addition of records with key values higher than the highest key value in the dataset, while the Extended Overflow blocks will handle all of the other file expansion needs.

For IAM files that are not SMS Extended Format, if they experience larger than anticipated growth, IAM can dynamically increase the size of the secondary space being requested. Once a dataset has five extents on any particular volume, IAM will increase the secondary space by the MAXSECONDARY factor, providing that the value does not exceed the original primary space requested, and that there is sufficient space on the volume for the desired quantity.

**Index Space**

The IAM Index Space is an area of virtual storage that IAM uses for to contain the indexes to the open IAM files. The Index Space can reside in 64-bit addressable virtual storage, or within a z/OS Data Space. Using the 64-bit addressable virtual storage eliminates the restriction of a maximum of 2 gigabytes for the index storage that exists for a data set that some users have encountered. IAM’s default is to use the 64-bit virtual if available, then a z/OS data space, and if neither of those are available then storage within the address space that is 31-bit addressable will be used. Users may need to code the MEMLIMIT JCL parameter on the EXEC to use 64-bit virtual storage.
Once processing has been completed for the IAM dataset, the application program should explicitly close the dataset. If the dataset is not explicitly closed, then task termination will close the dataset when the task that opened the dataset completes. The basic functions of close processing are to:

- On files opened for update, close processing will write out any updated blocks to the dataset that have not yet been written, and update the file statistics and control information.
- Close the internal DCB used by IAM to access the dataset.
- Freemain storage used to access the dataset.
- Delete the processing modules used for accessing the dataset.
- If an IAMINFO DD is present, format and produce the IAMINFO report.
- Produce an SMF record of file activity, if so enabled by the IAM Global Options.
- Disconnect the ACB from the control blocks built, and from the system.

Once the dataset is closed, no further I/O activity can take place. The application program should check the return code from the close to make sure that it was successful. A return code of 4 indicates that the close failed. It is rare that a close will fail, but it can happen. The main reason for a close to fail is that there is insufficient storage for the close to process. This is indicated by an error code of 136 (x'88'), that is returned in the ACB error code field. If this error occurs, the application should freemain some storage, or close other files, then reattempt to close the IAM file.

For IAM files opened for update, a file full error is not considered by IAM to be a fatal error. The Sx37 abends will be masked out by IAM for Enhanced Format IAM datasets, and treated as a file full logical error. The updated or inserted record will not be written out to the dataset. For updated records, the non-updated version of the record will remain in the dataset. The error conditions will be reported back to the calling program using standard VSAM logical error codes, and exits where applicable. To prevent getting into a reoccurring Sx37 loop, IAM will not try to take any additional extents for a file once such an error occurs, unless the dataset is closed and reopened.

File full errors can also occur when IAM is unable to obtain additional virtual storage for the expanding index structure. This type of file full condition is accompanied by an IAMW03 error message.
Special Considerations

JCL
In general, there are no JCL changes required to access IAM files in place of VSAM. The only required JCL parameters on the DD card for an IAM file are the DSN and DISP. If the JCL was set up to use Batch LSR for VSAM, that can be left in place for IAM, although IAM will not use any of the VSAM LSR buffers. Likewise, any AMP parameters can also be left in place. The BUFNI value will always be ignored by IAM, because it does not need or use index buffers. The BUFND value will be looked at and used as the value for MAXBUFNO, providing that there was no MAXBUFNO override, and that the value exceeds the default value for MAXBUFNO. STRNO will be used to establish the initial number of place holders, as done by VSAM.

While optional, users are highly encouraged to add an IAMINFO DD statement allocated to SYSOUT to job steps that are using IAM datasets. When this DD statement is present, a one-page report will be produced each time an IAM dataset is closed, providing file characteristics and statistics from the execution, including logical and physical I/O counts, and storage requirements. As an alternative, users are encouraged to set up the IAM Global Options to write out the IAMINFO as SMF records. These records contain the same information without the need for modifying JCL. The IAMINFO reports can then be produced by running the IAMSMF program’s IAMINFO command.

Multiple ACB
IAM datasets can be processed by multiple ACB’s opened within the same address space, either with the same or different DD names. IAM will share the control block structure and the index between the different ACB’s, as long as all of the ACB’s are opened under the same task.

The advantages of sharing the control block structure and index are that doing so significantly reduces the storage required to process the dataset with the multiple ACB’s, plus it provides for complete data integrity between the different ACB’s. If the sharing is not done, then each ACB is completely independent, and will be subject to the standard Cross Region sharing considerations. This means that unless a file is defined with Share Option 3 or 4, only one of the ACB’s can be opened for update.

Another advantage of the sharing is that for CICS, or other long running jobs, there is provision for 24-hour operation, if IAM RLS is not being used. This is achieved by having one of the ACB’s opened for read-only access, and the other being opened for update. When the update ACB is closed, all of the updated data buffers are written out, and the necessary control information in the dataset is updated. The dataset can then be updated by other jobs. Then, when the update ACB is reopened, all the buffers are invalidated to insure that only the current blocks from the dataset are used, plus the overflow and PE indexes are rebuilt to reflect the current state of the file. This will provide for online read only access to the file while it is being updated by batch processing, however the online system will not have full read integrity for updated, added, or deleted records.
IAM provides two features that enable sharing of IAM files at the record level. IAM/PLEX is an optional feature that enables applications running on different LPAR’s that are within a Sysplex to share IAM files for update processing with locking at the record level. IAM/RLS is the other feature and it is included with the base IAM product that enables applications running on the same LPAR to share IAM files for update processing with locking at the record level.

When using either of these two features the I/O requests are transferred over to an IAM/PLEX or IAM/RLS address space for processing, with data and status returned to the application program. Both of these features support concurrent batch and online system processing, along with journaling, recovery through the journals, and record locking. Each of the IAM/PLEX and IAM/RLS address spaces is assigned a unique 4-character identifier referred to as the RLSID. Each of the IAM datasets that is eligible for sharing is also assigned an RLSID to represent the address space that will handle that particular dataset. An RLSID is assigned to a dataset from the dataset inclusion/exclusion table, or by the RLSID CREATE or ACCESS override. The basic underlying rules are that a dataset can only be “owned” by one of either an IAM/RLS or IAM/PLEX address space, and that any particular job or CICS region can only be directly connected to one IAM/RLS or IAM/PLEX address space. With IAM/PLEX jobs can access files handled by other IAM/PLEX address spaces providing that they are assigned to the same RLS group.

For CICS, there are no changes to CICS application programs. There are some IAM provided CICS exits required so that IAM can properly identify and handle the CICS units of work.

Batch applications that do heavy update activity on recoverable files, that is those that have before image journaling enabled, will need to either use the automatic syncpoint capability, or include calls to the IAM syncpoint within their programs.

For applications that sequentially read a portion or the entire contents of an IAM file that is being processed through an IAM/RLS address space or a directly connected IAM/PLEX address space, a higher performance fast path will be automatically invoked. This fast path processing reduces the overhead caused by IAM/RLS, such that the sequential reader will have performance that is very close to the performance without using IAM/RLS. These results are achieved by IAM/RLS passing back several records to the job reading the file, thus significantly reducing the interaction between the job and IAM/RLS. The sequential fast path will be used only on sequential reads for input only. The sequential fast path is not used on sequential reads for update.

In test runs using the same 33-gigabyte file from the Long Key test, for an IDCAMS REPRO, the elapsed time for the RLS Fast Path run was 76% less than using RLS without the Fast Path. The CPU time was also reduced by 64%. This test was done using all default parameters.
The amount of data passed by IAM RLS to the application job can be controlled with the IAM ACCESS Override of RLSFP=. For this override, users can specify a value from 0 to 8. A specification of 0 will turn off the sequential fast path processing. Other values are used as multipliers on the file’s block size to determine the amount of storage that IAM RLS will use as a buffer between itself and the application program. The override is specified on the actual job step that is processing the IAM file sequentially. The default value is 2, meaning that approximately 2 blocks worth of data records will be passed back to the application program. For many jobs, the default value should provide sufficient benefit. For very large files, or files with long records, a higher value may improve performance even more. There is no IAM Global Option for the RLSFP value which can only be changed with an override.

**Journaling** IAM can also provide a journal for file update activity to facilitate recovery with the IAMJREST program. Journaling is activated based on either the JRNAD= override, or for DFSMS managed datasets by the LOG parameter on a DEFINE CLUSTER. Journaling can be performed either as a normal access function, or through IAM/RLS or IAM/PLEX. The normal access will have a separate journal for each IAM dataset, whereas the journals under IAM RLS contained the combined records from all datasets that are being journaled. For datasets that are not processed by IAM RLS, IAM RLS journaling be used by setting the IAM Global Option ENABLE=RLSJRN. Further information on journaling and recovery is in Section 10.88 of the IAM manual for normal access, or in Section 20 for IAM RLS access.
Overrides for IAM File Processing

IAM has an override facility that provides easy access to many of the IAM special features. Some of the more common reasons for using IAM Overrides include:

- Adjusting the range of buffers for IAM Real Time Tuning
- Placing the buffers in 64-bit addressable storage.
- Enabling use of the above the bar cache for overflow blocks with the CACHE64 override
- Enabling use of the IAM Dynamic Data Space or DYNCORE Dynamic Tabling feature
- Enabling use of IAM’s Index Space
- Enable or disable the IAM journaling
- Altering values for IAM’s Dynamic Region Adjustment
- Activating the BACKUPCOMPRESSED feature
- Altering IAM’s Dynamic Secondary Space adjustment feature values

How to Use Overrides

Use of the IAM Override facility is easy. Just add an IAMOVRID DD card to the job steps that are processing IAM datasets, and provide the simple control card image input with the desired parameters. The IAMOVRID DD is normally a DD * that is followed by the control cards within the input job stream. The IAMOVRID could also be a sequential or partitioned dataset member that contains the input control cards. The format of the override card for the ACCESS overrides are that they begin with the word ACCESS, which can begin in column one or be preceded by blanks, is followed by one or more blanks, then the override keywords with their values are specified, separated by commas. The keywords and their values may not go beyond column 71. Continuations, if needed, are indicated by leaving a comma after the last keyword on the preceding card, which is followed by other card(s) with the additional keywords. Full details on the IAM Overrides can be found in Section 30 of the IAM User’s Manual. If a dataset is going to be processed under IAMRLS, then for the job / address space actually using the dataset, the only relevant overrides are RLS, RLSFP, and RLSID, the rest of the overrides are ignored. Shown below are the commonly used IAM ACCESS Overrides that are applicable to IAM’s Enhanced Format Files.
Override Keywords

**Keyword** | **Brief Description**
--- | ---
BACKUPCOMPRESSED | Indicates that IAM is not to decompress the data when passing the requested records to the requester. The data will remain in an IAM data compressed format, which will not be usable by application programs for purposes other than copying and reloading an IAM data compressed dataset. **Backup compressed is only honored on software compressed files, otherwise it is ignored.** This option cannot be used with Syncsort.

BUFSPACE= | Specifies a value, in kilobytes, for that amount maximum amount of buffer space that should be used. Specification of the BUFSPACE override will supersede any MAXBUFNO specification.

BUF64= | Specifies whether or not IAM is to acquire buffers in 64-bit addressable virtual storage. Valid values are YES and NO. When 64-bit addressable buffers are being used, IAM acquires buffers in one megabyte increments, and fits as many buffers as it can within each allocated area. The default is based on the Global Options, which as shipped is NO. Use of 64-bit virtual storage for buffers may cause higher CPU times due to the overhead of page fixing that storage area.

CACHE64 | Keyword, with no operands, that specifies IAM is to use a 64-bit virtual storage area to hold an Enhanced format dataset's overflow blocks. The primary benefit is expected for programs accessing the file sequentially, when there are a lot of overflow records.

DDNAME= | Keyword that specifies the DD name(s) of the files to which the override values are to be applied. The special value &ALLDD can be used to apply the overrides to all files except those that are otherwise explicitly specified. Either DDNAME or DSN is required.

DSN= | Specifies the name of the dataset to which the override values are to be applied. Either DDNAME or DSN is required.

DYNCCORE=nnnn | Specifies the amount of virtual storage, in kilobytes, to be used for IAM’s Dynamic Tabling facility. Storage is acquired from above the line (31-bit addressable) storage within the job’s address space. This keyword cannot be specified if DYNDS is specified. Maximum value is 16000.

DYNDS=nnnn | Specifies the amount of virtual storage, in megabytes, to be acquired for a Data Space to be used for IAM’s Dynamic Tabling function. Values from 1 to 2048 are permitted. Cannot be specified with the DYNCCORE override.

INDEXSPACE= | Indicates where IAM will acquire the virtual storage for the index. Valid values are:

  - **64BIT** - Storage is acquired in 64-bit addressable storage in region.
  - **YES** - Storage is acquired from a z/OS data space
  - **NO** - Storage is acquired from 31-bit addressable storage in region.

Default value is 64BIT, unless otherwise changed in the IAM Global Options.
**Keyword** | **Brief Description**
--- | ---
JRNAD= | Specifies the IAM journaling capabilities to be used. When specified on the ACCESS override statement, if the value specified enables journaling, the value will be combined with any specification from the CREATE override when the file was defined or loaded.

For files not being processed by IAM/RLS or IAM/PLEX, users must pre-allocate and catalog a log dataset to be used for the journaling, which is required to be the name of the IAM dataset appended with the characters ".LOG". IAM journaling will not be active during file loads, reorganizations, or during recovery from the journal.

For files being processed by IAM/RLS, the users must specify the journal datasets to the IAM/RLS startup procedure if they want to use journaling, and had not specified to do so when the dataset was defined or loaded.

The ACCESS override only changes the journaling value for the job step on which it is specified. If the dataset is being opened under IAMRLS, then the JRNAD override MUST be specified under IAMRLS, and not the batch jobs using IAMRLS services. Valid values are:

**BOTH or ALL** - The IAM log dataset will contain both before and after images. This will enable the user to perform either a forward recovery, or a backward (backout) type of recovery.

**BEFORE** - The IAM log dataset will contain before images of updated records. This option allows backward (backout) recoveries only. If AFTER images were specified on a CREATE override, then AFTER images will still be logged to the journal.

**AFTER** - The IAM log dataset will contain after images of updated records. This option allows forward recoveries only. If BEFORE images were specified on a CREATE override they will still be logged to the journal.

**NONE** - The IAM journaling feature will not be used for this IAM dataset. This will turn off any journaling for this job step, even if journaling had been specified on a CREATE override.

Default value is NONE, unless otherwise specified when the file was defined or loaded.

**MAXBUFNO=nnnnn** | Specifies the maximum number of buffers IAM will use for the specified file. Maximum value allowed is 24576.

**MAXREGION=nnnnn** | Specifies the maximum value, in megabytes, which IAM Dynamic Region Adjustment will allow the Extended Private region to be set to. Maximum value allowed is 2048.

**MAXSECONDARY=nn** | This applies only to datasets that are not SMS Extended format. Specifies the multiplication factor to be used on the secondary quantity when a dataset exceeds 5 extents on a volume. Maximum value that can be specified is 10. Note that IAM will never increase the secondary space to a value greater than the primary space.
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINBUFNO=n</td>
<td>Specifies the minimum number of buffers that IAM is to use for the specified</td>
</tr>
<tr>
<td></td>
<td>dataset. If specified, this will also be the number of buffers that IAM initially</td>
</tr>
<tr>
<td></td>
<td>acquires during open.</td>
</tr>
<tr>
<td>NORLS</td>
<td>A keyword that specifies that IAM is not to process this file under the IAM RLS</td>
</tr>
<tr>
<td></td>
<td>address space, even if the installation's RLS selection criteria indicates that</td>
</tr>
<tr>
<td></td>
<td>the file(s) are eligible for RLS processing.</td>
</tr>
<tr>
<td>REREAD</td>
<td>Indicates that IAM will always reread the overrides each time an IAM file is</td>
</tr>
<tr>
<td></td>
<td>opened. Normally for batch jobs, IAM will only read the overrides on the first</td>
</tr>
<tr>
<td></td>
<td>open, and save all the values in virtual storage for subsequent reference.</td>
</tr>
<tr>
<td></td>
<td>Specifying this keyword allows the overrides to be changed for long running</td>
</tr>
<tr>
<td></td>
<td>online systems. Reread is the default for CICS, IAM/RLS, and IAM/PLEX address</td>
</tr>
<tr>
<td></td>
<td>spaces,</td>
</tr>
<tr>
<td>RLS</td>
<td>A keyword that when specified indicates that the file(s) are to be processed</td>
</tr>
<tr>
<td></td>
<td>by the IAM RLS address space. If the IAM RLS address space is not active, the</td>
</tr>
<tr>
<td></td>
<td>OPEN of the file will fail.</td>
</tr>
<tr>
<td>RLSFP=n</td>
<td>When a file is being sequentially read for non-update and is being processed</td>
</tr>
<tr>
<td></td>
<td>under IAM RLS, this parameter controls the amount of data that is passed back to</td>
</tr>
<tr>
<td></td>
<td>the calling address space. Non-zero values offer a sequential fast path function</td>
</tr>
<tr>
<td></td>
<td>that will reduce the overhead of using IAM RLS. The value specified is a multiplier</td>
</tr>
<tr>
<td></td>
<td>times the block size of the data set. Specification of 0 will turn off the fast</td>
</tr>
<tr>
<td></td>
<td>path function, which will guarantee that an application will always get the latest</td>
</tr>
<tr>
<td></td>
<td>copy of the record. The default is 2, meaning approximately 2 blocks worth of</td>
</tr>
<tr>
<td></td>
<td>data will be passed back to the calling address space.</td>
</tr>
<tr>
<td>RLSID=n</td>
<td>Specifies the four character identifier of the IAMRLS address space that is to</td>
</tr>
<tr>
<td></td>
<td>be connected to. This keyword is primarily intended to be used with a DD=&amp;ALLDD</td>
</tr>
<tr>
<td></td>
<td>override card, and then with any other override cards that are specified for that</td>
</tr>
<tr>
<td></td>
<td>job step. Specification of an RLSID on the ACCESS override does not force a</td>
</tr>
<tr>
<td></td>
<td>dataset to be processed under IAM RLS, but rather directs it to an IAMRLS address</td>
</tr>
<tr>
<td></td>
<td>space if IAM RLS processing is otherwise indicated for this dataset.</td>
</tr>
<tr>
<td>SHAREOPTION=n</td>
<td>Overrides the VSAM Cross Region Share Option to use for this execution. Provided</td>
</tr>
<tr>
<td></td>
<td>as the alternative to performing an ALTER on the file to change the Shareoptions.</td>
</tr>
<tr>
<td>TURBO=[YES</td>
<td>NO]</td>
</tr>
<tr>
<td></td>
<td>that IAM will be more aggressive in increasing the number of buffers. Default</td>
</tr>
<tr>
<td></td>
<td>value is from the IAM Global Options Table, which as shipped is YES.</td>
</tr>
</tbody>
</table>
Examples of Accessing an IAM Dataset

A few JCL examples of processing an IAM dataset are shown below. Essentially, there are no JCL changes to use an IAM dataset instead of a VSAM dataset, except when one desires to obtain the IAMINFO report, or wants to provide some IAM Overrides.

This example demonstrates the simplicity of using IAM. The only change made to this job stream when converting to VSAM was the optional addition of an IAMINFO DD card, to obtain the IAM run time reports. Note that the VSAM AMP parameter has been specified for one of the IAM datasets. The BUFNI specification is ignored by IAM, however the BUFND value may be used, if it is higher than the default for MAXBUFNO.

```
//PROCESS EXEC PGM=anypgm
//SYSPRINT DD SYSOUT=* 
//FILEA    DD DSN=prod.iam.filea,DISP=SHR
//FILEB    DD DSN=prod.iam.fileb,DISP=SHR
//FILEC    DD DSN=prod.vsam.filec,DISP=SHR
//MASTER   DD DSN=prod.iam.master.file,DISP=SHR, 
//          AMP=('BUFNI=6', 'BUFND=60')
//IAMINFO DD SYSOUT=* 
```

Figure 10-31: Example of Processing an IAM Dataset (EX1040A)
Example B: Using IAM ACCESS Overrides

This example demonstrates how to use the IAM Override facility when processing multiple IAM datasets. Because the job is processing several datasets, some of them with large indexes, the INDEXSPACE=YES override is being specified with DD=&ALLDD, so that the virtual storage for the index will not come out of the job’s address space.

The TABLE file is a read only reference KSDS dataset that is used for data validation. The records in this dataset tend to be on the small side, less than 100 bytes, and processing is random against a subset of the records, which tend to be referenced many times. For this dataset, the IAM Dynamic Table Facility is quite useful. An IAM Override is being used to enable IAM's Dynamic Tabling Facility, with a Data Space being requested of 32 megabytes. The INDEXSPACE override is also being specified for this dataset, because it is being explicitly overridden and the value will not be picked up from the DD=&ALLDD override card.

```
//MASTRUPD EXEC PGM=updpgm
//SYSPRINT DD   SYSOUT=* 
//MASTER   DD   DISP=SHR,DSN=prod.master.iam.file
//SEQFILE DD   DISP=OLD,DSN=prod.seqfile.iam.esds
//TABLE    DD   DISP=SHR,DSN=prod.table.iam.file
//FILEA    DD DSN=prod.iam.filea,DISP=SHR
//FILEB    DD DSN=prod.iam.fileb,DISP=SHR
//FILEC    DD DSN=prod.iam.filec,DISP=SHR
//IAMOVRID DD   *
   ACCESS DD=TABLE,DYNDS=32,INDEXSPACE=YES
   ACCESS DD=&ALLDD,INDEXSPACE=YES
 /*
 //IAMINFO DD   SYSOUT=* 
```

Figure 10-32: Example of Processing IAM Datasets with Overrides (EX1040B)
IAM and CICS
IAM is designed to provide a transparent high performance disk file manager alternative for VSAM KSDS, ESDS, AIX and RRDS files. IAM can eliminate VSAM performance bottlenecks and reduce VSAM file size (DASD footprint) by as much as 50%. This makes IAM an important performance enhancement option within an online transaction processing environment such as within CICS.

In a CICS environment IAM improves transaction response time by reducing the time taken for access to VSAM files. Files, including alternative indexes and paths, that have been converted from VSAM to IAM may be used by a CICS TS region without any changes to the CICS startup JCL, to the RDO file definitions within CICS, to the CICS start up parameters, or to any of the IAM default global options.

To effect this improvement that IAM provides and to insure optimal performance when using IAM there are several areas within CICS that need to be taken in consideration when implementing IAM files as listed below:

The primary areas that are addressed in this section include:

- IAM VSAM Interface (VIF) start order
- IAM Global Option BELOWPOOL
- CICS System Initialization Table CILOCK parameter
- IAM Statistics and Performance Reports - IAMINFO
- Effective use of IAM Override ACCESS Statements within CICS
- IAM DEFERWRITE Override option
- CICS and IAM Buffer Management and Storage Utilization
- CICS and IAM use of VSAM Strings
- CICS NSR and LSR specifications
- CICS and IAM with the BUFND (or DATABUFFERS in RDO) setting
- IAM index storage
- IAM Dynamic Data Space
- IAM and 64 Bit Above the Bar CACHE64 storage option
- IAM with the CICS RDO FIRSTREF FILE Open option
- CICS Concurrent VSAM sub-tasking option
- CICS Open Transaction Environment (OTE) THREADSAFE/OPENAPI options
- IAM with the CICS RDO FIRSTREF FILE Open option
- CICS Shutdown processing and IAM file closings

IAM'S VSAM INTERFACE (VIF)
The IAM VSAM Interface (VIF) for the level of IAM that CICS will use must be active when CICS is started. If a CICS region that accesses IAM files is started before the VIF, that region must be recycled before it will be able to process IAM files.
**BELOWPOOL**

The IAM Global Option Table **BELOWPOOL** option controls whether IAM will use (or not use) the below the line storage pools for I/O control blocks within a CICS or an IAM RLS address space. The default for this option is **BELOWPOOL=YES**. The use of the below the line storage pools substantially reduces IAM’s use of 24-bit addressable storage. If the default is changed or overridden to **BELOWPOOL=NO** then IAM’s will use a minimum of 4K below the line storage area for each open IAM file. Hence, within a CICS or an IAM RLS region that has many IAM files opened concurrently then leaving this option set to **BELOWPOOL=YES** will reduce the below the line storage requirements by at least 50% when several IAM files are opened.

**CILOCK**

The CICS System Initialization Table (SIT) parameter CILOCK defaults to NO. This setting is intended to reduce the number of VSAM exclusive control conflicts when reading VSAM files for update. It does so by using extra CPU cycles and doing extra I/O. IAM locks at the record level, not at the control interval level, thereby eliminating this delay except when concurrent transactions are attempting to update the same records. Therefore the CILOCK parameter should be set to **YES** when most or all of the VSAM files have been converted to IAM.

If CICS is running with the default CILOCK value of NO and IAM files are being shared with IAM/PLEX or IAM/RLS, then it is critical to provide a sufficient value for the VSAM STRINGS. Failure to do so can result in CICS deadlock situation if a waiting for strings condition occurs. See the section below on VSAM Strings for additional information.

**IAMINFO Reporting**

The IAMINFO DD card can be added to the JCL for any CICS region that uses any IAM file(s). The actual output class should be set to a held class that is deleted once it is a month old. This will ensure that a history of the activity against IAM file(s) is available should any performance issues arise. This output should be scanned for informational and warning messages on a regular basis. An example of the DD card is:

```
//IAMINFO DD SYSOUT=*
```

Where “*” is any valid JES output class

The addition of an IAMINFO DD card to the CICS JCL is normally recommended to obtain the IAMINFO reports for each IAM file that is accessed from CICS. However, if you have a large number of IAM files it is recommended instead that IAMINFO reports be generated by using the IAM SMF option. Doing this will avoid having IAM’s generation of the IAMINFO reports causing an elongation of the time required for performing an expedient CICS shutdown.

Generating a large number of IAMINFO reports to the IAMINFO DD at CICS SHUTDOWN can introduce a delay in CICS shutdown processing just to format and generate the reports. Changes have been made for IAM Version 9.0 to reduce the delay significantly, but it is not completely eliminated. Using the IAM SMF option would only require the collection of IAM SMF data records which is of minimal overhead. It is from the individual IAMINFO reports for each IAM file, one can determine the buffer and storage utilization during a specific CICS run. Note that an IAMINFO report is generated each time an IAM file is closed (or an IAM SMF record). There can be several individual reports (or IAM SMF records), if an IAM file is opened and closed multiple times during the lifetime of a CICS region.
Instructions and examples on using the IAMSMF program to produce the IAMINFO reports from IAM SMF records are in Section 10.72 and Section 41.04 of this manual. Instructions for setting IAM options to produce the SMF records are in Section 91. Be sure that your procedures to offload the SMF records includes saving the SMF record number selected for the IAM records.

IAM Overrides

IAM overrides can be used with CICS regions. To do so an IAMOVRID DD card is added to the JCL for the CICS region(s) that are going to have IAM overrides. There is no FCT entry required for the IAMOVRID file. The IAMOVRID DD should specify a physical sequential card image file or to a card image member in a partitioned data set. For example:

```
//IAMOVRID DD DISP=SHR,DSN=my.cics.iam.override(cics123)
```

With IAM Version 9.2 the overrides are automatically reread for each IAM file open, so the REREAD keyword is no longer necessary.

An example of some IAM override cards that could be in the IAMOVRID dataset are:

```
ACCESS DD=file1,BUFSPACE=2048
ACCESS DD=file2,BUFSPACE=32768,BUF64=YES
ACCESS DD=file2,DYNDS=1024
```

DEFERWRITE Override

In a CICS region IAM will immediately write out any block that contains a randomly updated record. The effect of this is that with each update, insert, or delete record request will generally cause one physical WRITE out of a block to the dataset. The exception to that are when a mass delete of records operation is performed, or when the writes are performed with MASSINSERT specified, as those are sequential requests. Sequential updates are generally deferred.

If the data contained within the dataset is not highly critical or time sensitive then performance may be improved by specification of an override (in the IAMOVRID DD) to CICS for those specific IAM file(s) with the parameter DEFERWRITE=YES. The following is an example (using the abbreviated “DEFERW” for the DEFERWRITE keyword) of this parameter:

```
ACCESS DSN=prod.$iam.trans.log.file,DEFERW=YES
```

This will cause IAM to delay the writes until the buffer is needed for a different block. It is important to note the implication of data integrity with specifying DEFERWRITE=YES. When the writing out to DASD of updated data blocks is deferred, then it is possible for the updated records to not get hardened to DASD for an extended indeterminate period of time. The cautionary note is that if the system crashes, or the job is forced out, then IAM is precluded from properly closing the file (flushing the buffers to DASD). The result is potential for data (added, updated or deleted records in the blocks not written out) to be lost. The records could potentially be recovered if they are being also written out to a journal.

As such it is recommended that specifying DEFERWRITE=YES should be utilized with caution in an online environment such as with CICS.
Buffer management for IAM is performed outside of the sphere of both VSAM and CICS. This allows IAM to dynamically adjust the buffer management techniques based on the actual type of I/O being performed by the application. This eliminates the tuning decisions of NSR or LSR and the number of index buffers. There are no index component buffers used by IAM, as the index is always read into virtual storage when the file is opened. IAM’s Real Time Tuning will adjust the number of buffers being used by a dataset automatically as needed, based on default limits set by the installation (in the IAM Global Options Table), or based on any provided overrides in the “//IAMOVRID DD.” file.

The IAM CICSBUFSP=nnnnn parameter defines the default maximum amount of storage that IAM is to use within a CICS region for allocation of its buffers per opened IAM file. The default value is 1024, in K byte increments that IAM will attempt to utilize for buffers for files opened under a CICS region. As CICS regions vary considerably it is advisable that customers should carefully evaluate this default value, and revise it either higher or lower as is appropriate for their environment and requirements.

When a lower value is specified for CICSBUFSP, customers will have to provide individual IAM overrides for the busiest files. These individual overrides for the busiest files will allow for a sufficient number of buffers to meet their needs of the CICS applications accessing the files. The default value for CICSBUFSP in IAM is set to allow for sufficient buffers for the majority of the files that would need specific IAM overrides for buffering and thus make the IAM product easier to use.

The formula for initially estimating the maximum storage that could be used by IAM within a CICS region for buffers is simply to multiply the CICSBUFSP value times the anticipated number of concurrently open IAM datasets. This is then the maximum amount of storage that IAM will utilize for buffers in a CICS region. It is important that this not constrain the CICS regions over all utilization of storage, hence it is imperative that a careful monitoring of file activity and storage utilization be done on a regular and ongoing basis.

An appropriate default value for any particular CICS region is going to vary depending on the number of IAM files within the region, the virtual storage limitations within the CICS region, the REGION value specified on the JCL, the volume and type (READs, WRITES, etc.) of IAM file I/O activity, and the response time objectives of the CICS region. Often it is found that it is more efficient utilization of storage to specify individual BUFSP or MAXBUFNO override values for selected high activity IAM files and reduce the default CICSBUFSP value for the remaining files to a lower amount. The following is an example of specifying the BUFSPACE parameter for a single dataset (using the abbreviated BUFSP keyword) and the MAXBUFNO for a different dataset (using the abbreviated MAXB keyword):

```
ACCESS   DSN=PROD.$IAM.TRANS.LOG.FILE,BUFSP=1024
ACCESS   DSN=PROD.$IAM.MSTR.ACCTS.FILE,MAXB=2048
```

The recommendation for number of strings is the same as it is for VSAM. Basically the value for number of strings to specify is the maximum number of concurrent transactions that are active and browsing or updating a particular dataset. It is also advisable to increase this value by a few extra strings (according to CICS documentation that is 20%) to accommodate any general random reads that could be concurrently active against the dataset, so as to avoid string waits.
IAM behaves the same way VSAM does regarding strings. A string holds the position in a file as long as a position type request is issued. CICS monitors string usage (either at the file level for non-LSR, or at the LSR buffer pool level), and if all of the strings (place holders) are in use, CICS puts the requesting transaction in a string wait, until a place holder becomes available. For non-LSR this is done to prevent VSAM from acquiring more storage for strings, and for LSR it is done to prevent VSAM from failing the request due to insufficient strings.

If an installation does not specify an adequate number of strings to support its concurrent I/O activity, then string waits can occur resulting in a negative impact on response times. The vast majority of string waits that occur are because an installation has specified an inadequate number of strings to handle their processing demands. Things that affect the number of strings needed include the I/O device response time, the amount of buffering being done (more buffers reduce physical I/O, therefore making I/O requests run faster and hence normally reduced need for strings), and the amount of browsing or updating being done to a file.

String numbers are not all that important to IAM itself, as IAM will acquire additional strings as needed, however CICS monitors string utilization and will internally delay I/O to a file if CICS believes that all the strings are in use. So proper setting of STRNO is important to prevent CICS from delaying I/O requests.

Some additional notes on Strings:

1. When using IAM/PLEX or IAM/RLS and the default CILOCK value of NO, it is critical that a sufficient number of strings be specified. With CILOCK=NO, CICS will cause a transaction that is holding a record lock to release a string. If other transactions require the same record lock, they will go into a wait for the record lock while holding their string. If no strings are available when the owning transaction is ready to update the record then a deadlock condition occurs. The recommendation is to run with a CILOCK value of YES or provide a sting value that exceeds the maximum task value to ensure there will never be a string wait.

2. CICS will reserve a proportion (20%) of the STRINGS value to be used for read-only requests. Hence tasks in CICS may go into a string wait situation while the total tasks currently active against a file is less than the total number of STRINGS specified.

3. In setting the STRINGS value for an ESDS file, consider that if the ESDS is a log file, where it only gets records "added" (WRITE I/O operation) then set STRINGS to 1 (one). If a STRINGS value were set to be greater than 1, it can lead to exclusive control conflicts where other tasks may be attempting to write to the ESDS file at the same time. A STRINGS value of 1 (one) will preclude this as CICS (via the CICS dispatcher) will then enforce serialization and single threading of activity to the file.

For IAM ESDS files, if the order of records from concurrently running transactions can be intermixed, then consider setting the IAM Global Option DISABLE=ESDSLOCK and using a higher value for number of STRINGS. These changes will further enhance the performance of transactions using IAM ESDS files.

4. The use of LSR buffering has an impact on how CICS monitors STRINGS. The effect is that there are two limits on strings: one for the individual file and then also the limit on strings for the total LSR pool to which the file is defined. In general, the actual number of strings acquired by CICS for an LSR pool will be less than the cumulative total for all of the datasets in that pool, unless STRINGS is explicitly specified with a larger value on the LSR pool definition. In
a particular LSR pool, one could run out of strings before the actual limit for any particular file in the pool was reached. This is a CICS arbitrary mechanism for controlling tasks and file access within CICS, not a VSAM or IAM attribute. While IAM does not use any of the strings for the LSR pool, CICS thinks that it does and includes IAM files in the string usage for an LSR pool.

**NSR or LSR**

IAM’s Real Time Tuning buffer management techniques are automatically used for all IAM files, so the specification of the type of buffering has more of an impact on how CICS is interfacing with IAM than with any actual influence on IAM buffers or buffer management. The primary difference is how asynchronous processing is achieved such that the access method will not go into a wait.

With NSR, the typical asynchronous mode is used where CICS will issue WAIT’s completely under its control, and when the I/O request is posted complete CICS will then issue the CHECK macro. Another distinction of NSR buffering is that the buffers are actually obtained by VSAM itself. So when IAM is being used, IAM will acquire the buffers and there are no VSAM buffers obtained.

With LSR buffering asynchronous processing is achieved through use of the VSAM UPAD exit, in which whenever the access method (IAM or VSAM) needs to wait it will give control back to CICS through an exit mechanism. The UPAD exit mechanism provides CICS with full control of when VSAM and IAM processing actually occurs and is more efficient than the typical asynchronous processing. With LSR buffering, CICS has to invoke a VSAM service to build the buffer pool. IAM does not utilize any of the buffers in the LSR pool, so the down side is that if LSR buffering is specified there may be some under utilized buffers that could be using up some valuable virtual storage.

General recommendations are when first converting from VSAM to IAM leave the buffering specification as it is with VSAM. Subsequently once IAM is proven then consider revising the buffering to minimize impact on the most constrained resource. If virtual storage is constrained, then either use NSR buffering, or manually reduce the amount of buffers acquired by CICS for the LSR pool. If CPU time is the most constrained, then use LSR reducing buffers for the LSR pools as appropriate.

One other consideration is that files that are in Data Tables must be in the LSR pool. IAM does provide the Dynamic Data Space function that may for some applications be a good alternative to the CICS Data Tables, and can be used with either NSR or LSR buffering.

**Data Buffers (BUFND)**

The BUFND parameter is usually found in JCL as an option on the AMP keyword. In general, there are no JCL changes required to access IAM files in place of VSAM, no matter whether it is a batch job or underneath a CICS region. The only required JCL parameters on the DD card for an IAM file are the DSN and DISP. In CICS region, if the files are typically defined in the CICS DFHCS0 file via RDO and nowadays dynamically allocated to the CICS region without having them defined in the CICS startup JCL.

The corresponding BUFND parameter for the CICS RDO File Definition is the DATABUFFERS attribute. (Note the INDEXBUFFERS attribute, which is equivalent to the BUFNI in the AMP option of a DD JCL statement, will always be ignored by IAM. This is because IAM does not need or use index buffers.) If there are no MAXBUFNO or BUFSPACE overrides for the dataset, IAM will set the maximum number of buffers (MAXBUFNO) to the value specified by the BUFND (DATABUFFERS) if it is greater than the default value for maximum buffers calculated from the IAM Global Options table.
For CICS files not in a LSR pool, the value for the DATABUFFERS attribute of a CICS File Definition in RDO will result in changing the BUFND value in the ACB that is subsequently used by VSAM.

CICS regions both historically and currently are often beset with storage constraint issues. Most every resource within a CICS region takes up some sort of storage “footprint”. Hence, the use of storage within a CICS region is one critical resource that needs to be monitored attentively and periodically as the composition of the activity within a CICS region changes over time.

The IAM product provides the means to efficiently utilize CICS storage resources for IAM files. One of the means of significantly reducing the IAM storage footprint within a CICS address space is where the index to open files is kept. The INDEXSPACE option allows for large amounts of storage that might have been utilized within the CICS address space to contain IAM file indexes to be offloaded to separate z/OS storage areas that reduce storage contention. The index storage can be from either storage above the 31-bit addressable storage area known as 64-bit addressable virtual storage, or from a z/OS data space. The area selected is called the IAM Index Space, with default values set in the IAM Global Options INDEXSPACE or with the INDEXSPACE IAM override.

The INDEXSPACE=64BIT is the default option as IAM is shipped and that applies to CICS regions. When the first local IAM dataset is opened under CICS, IAM checks to see if there is at least 1024 megabytes available, based on the DATASPACE Global Option value that defaults to 1024 megabytes. If there is sufficient 64-bit storage available, then IAM will use that storage. To help make sure that 64-bit addressable storage is used users can specify a MEMLIMIT on wither the EXEC card, or in the active SMFPRMxx member in parmlib.

If there is not sufficient 64-bit virtual storage IAM will attempt to acquire a z/OS data space. The size of the Data Space requested for the Index Space is taken from the IAM Global Options Table, using the value specified for DATASPACE. Note, that this is the same value that is used for the data space obtained for a file load. The z/OS Data Space that contains the Index Space is initially created to be extendable, so it can be expanded in size as needed, with the maximum size set to four times the specified DATASPACE value (up to the 2 gigabyte limit).

The default value for the data space size is 1024 megabytes. The IAMINFO report generated at file CLOSE contains information on the data space usage for the particular dataset, and to the total data space usage for the job (or CICS region). It is recommended that these values be monitored, in the event that the value for the default data space size needs to be increased.
DYNDS  The IAM Dynamic Data Space \((\text{DYNDS}=\text{nnnn})\) feature provides additional benefit in a CICS online environment by using a z/OS Data Space for the caching of random accessed records from an IAM dataset. This is implemented as an enhancement to the Dynamic Tabling option of IAM with the table now contained within a separate z/OS Data Space. The Dynamic Tabling is a special feature that keeps the most frequently referenced randomly read records in a virtual storage cache. As Dynamic Tabling is record oriented, and is most useful when a subset of the records scattered throughout the file are repeatedly referenced, potentially reducing the virtual storage needs to have a large number of buffers. This Dynamic Data Space provides substantial benefit particularly for some online applications, by eliminating physical I/O as records are repeatedly retrieved from the table. Backing the Dynamic Table with a z/OS Data Space allows the dynamic table to contain a larger number of records (that can be contained within a maximum \(\text{DYNDS}=2048\), which is a 2 gigabyte Data Space).

IAM uses algorithms to manage the storage used within the Data Space to maximize the effective utilization of the storage resources and provide for efficient record search with reduced CPU overhead when inserting and removing records from the Dynamic Table in the Data Space. IAM utilizes a true \textit{least recently used} algorithm for record selection when IAM is required to make room for more recently referenced records. The use of the Dynamic Data Space is on a file by file basis, with each individual data set for which the function is requested using its own data space.

The capability is easily implemented by the use of the IAM ACCESS Override of \textit{DYNDS}. Using the \textit{“DYNDS=“} override, users simply specify the amount of storage that IAM is to use for the Dynamic Data Space in megabyte increments, up to a maximum of 2048, or 2 gigabytes. The virtual storage required for management of the Dynamic Data Space will come out of the Data Space itself so that use of the Dynamic Data Space will not impact virtual storage resources within the CICS address space.

The following is an example of using the DYNS option for a single IAM dataset in which a data space of 1024 megabytes will be created:

\[
\text{ACCESS DSN=PROD.$IAM.MSTR.ACCTS,DYNDS=1024}
\]

Monitor the paging statistics for the CICS address space. The use of additional z/OS data spaces to hold records and buffers in memory comes at a cost, where the cost is a potential increase in paging. This additional paging that can be attributed to the CICS address space, will slow down the CICS address space from being dispatched from a z/OS perspective. Paging operations cause a z/OS address space to stop while the “page in” operation occurs. That is a definite potential negative impact upon an online system such as CICS. Conversely, the positive impact will be with \textit{DYNDS} of having randomly read records available within memory which translates to reduced response times within a CICS environment.

CACHE64  The \textit{CACHE64} IAM access override option indicates that IAM should utilize 64-bit addressable storage as a cache area for Enhanced-format IAM files to contain the file’s extended overflow blocks. It is generally not recommended that the \textit{CACHE64} options be utilized within an online processing environment such as CICS. The \textit{CACHE64} option is better suited for batch jobs that process IAM datasets with large overflow areas, especially when they are being read sequentially. For randomly read IAM datasets as is more typically the method of access performed underneath CICS, the potential benefit of \textit{CACHE64} is minimal. Hence it is not generally recommended for CICS regions.
The CICS initialization parameter VSP=1, controls the attaching of a concurrent TCB within the CICS region. This concurrent TCB is utilized by CICS to offload only those File Control VSAM WRITE I/O requests onto a separate subtask. This is known as the "CO" (Concurrent) TCB. The "normal" or traditional CICS subtask is the "QR" (quasi-reentrant) TCB. It is on the QR TCB that the remaining I/O operations are performed.

The support for this was originally put into CICS to prevent any VSAM CI/CA split activity that may result from a WRITE I/O operation from holding up the dispatch of transactions on the QR TCB. With IAM, there is no CI/CA split activity that occurs as in traditional VSAM, hence there are no special considerations. IAM fully supports I/O requests from concurrent multiple subtasks.

The ability of CICS File Control I/O requests to be processed underneath a separate MVS TCB other than the traditional CICS "QR" TCB (Quasi-Reentrant TCB) was introduced in CICS TS V3.1 level when CICS File Control was allowed to be “Threadsafe”. This allows IAM’s individual I/O requests to be executed now underneath CICS TCBs known as L8 or L9 TCB. In environments with transactions accessing both DB2 and VSAM (IAM) there is no requirement or need to incur a task switch back to the QR TCB by CICS to perform the VSAM(IAM) File control requests. The transaction remains on the OPENAPI TCBs of L8 or L9 after having been switched to an L8/L9 TCB when the first DB2 request was issued. This reduces the instruction path by 2000 instructions or so for each direction of a task switch within CICS. (Going from the single QR TCB to L8/L9 TCB or back form an L8/L9 back to the single QR TCB costs approximately 2000 instructions per switch.)

All the IAM programs (whether they are running as CICS GLUE or TRUE exits within CICS or as lower level Access Method code) are programmed to be “THREADSAFE” per CICS requirements.

This multiple CICS TCB sub-tasking of application programs using the OPENAPI option to enable a program to run on a non-QR TCB, can cause some discrepancies in attributing and correlating I/O overhead (specifically IAM I/O response) back to a specific z/OS TCB within the CICS address space. The I/O overhead of accessing an IAM dataset will largely be attributable to the QR TCB except for the previously mentioned WRITE requests that are attributable to the CO TCB. Additionally, in the event that IAM is running underneath the CICS TS V3.2 and higher levels where the File Control Domain VSAM activity is running as “Threadsafe”, then in such situations IAM activity can run concurrently on multiple OPENAPI TCBs (L8 or L9 TCBs). The net effect is that I/O overhead (IAM I/O response) is then distributed across those OPENAPI TCBs underneath which IAM I/O requests ran.

Defining an IAM file to CICS via the RDO (Resource Definition Online) access to the CICS definitions file “DFHCSD” requires only that an IAM file be defined the same as a traditional VSAM file (ESDS, KSDS, RRDS or and AIX). There are though some implications that specifying select options with an IAM file will cause different behavior than with traditional VSAM. This includes the CICS parameter:

```
OPENTIME ([FIRSTREF | STARTUP])
```

This parameter specifies when a file is to be opened in a CICS region.
Implications of OPENTIME Options

When an IAM file is opened underneath CICS (without IAMRLS) IAM will build its internal index structure. This necessitates a complete full scan of the file to obtain all the relevant index values. The index structure is placed in the INDEXSPACE (which is an IAM managed area of virtual storage either in 64-bit addressable storage or a z/OS data space) attached to the CICS region. The result is if STARTUP is specified for all files, this will cause an elongation of CICS startup while IAM performs its index structure build function. However, if “FIRSTREF” is specified than this index build overhead/delay will be incurred by the first transaction that accesses or references the IAM file, with minimal delay during CICS initialization.

Additionally, if IAM files are repetitiously OPENED and CLOSED, and subsequently REOPENED within CICS then the overhead of the index build can become noticeable as the interval between OPENS/CLOSES decreases. There are some packaged applications that have common I/O routines that always issue an EXEC CICS OPEN and CLOSE requests bracketed around the access requests (READ, WRITE, etc) to a VSAM file. In CICS this can cause unnecessary overhead and delay when the accessed file is an IAM file, as the in memory (or INDEXSPACE Data Space) internal index structure is rebuilt each time the file is OPENED.

CICS Shutdown

Under CICS the OPEN and CLOSE processing occurs underneath the FO (File owning TCB). The result is that this can become a serially threaded bottleneck at CICS shutdown as all the CLOSE requests are being processed. If the default CICS shutdown process is left intact then there is the possibility that S33E abends can occur. This can occur when the "NORMAL" CICS shutdown request is issued. Typically this is done via the "CEMT PERFORM SHUTDOWN" request. The default action of this command in CICS is to cause the invocation of the IBM CICS supplied CESD transaction.

The transaction "CESD" defaults to invoking the IBM supplied program "DFHCESD". This program is one of the “User Replaceable Modules” (URM) within the CICS TS product. The DFHCESD program is supplied in both source and executable formats. Additionally IBM supplies alternative COBOL and PL/I sources (DFH0CESD and DFH$CESD respectively). These sources are in the CICS “SDFHSAMP” libraries. Within the program(s) the specified default delay between the initial CEMT PERFORM SHUTDOWN command’s invocation of this shutdown assist program DFHCESD via the CESD transaction and subsequent "re-invocation" of the same transaction/program by an internally issued START TRANSACTION(CESD) DELAY(hhmss) command is 120 seconds (two minutes).

On the re-invocation the Shutdown Assist Program (DFHCESD, DFH$CESD or DFH0CESD) will issue internally an "IMMEDIATE" shutdown. (via an "EXEC CICS PERFORM SHUTDOWN IMMEDIATE" program interface call.)

Since this CICS Shutdown Assist program is one of the "User Replaceable Modules" within CICS, it is designed to be modified by the user. The modified source program can be reassembled (or recompiled if using the COBOL or PLI versions) to generate a new executable module. Thus, the user can change the default DELAY value within the program to whatever value is appropriate within their CICS environment to insure a clean and orderly shutdown.

If the user’s CICS regions are such that they have numerous VSAM and IAM files to close and that the CICS region is not getting sufficient z/OS dispatch or CPU cycles, they will potentially incur the situations where a file close request doesn’t complete on a subtask before the CICS main task unilaterally issues a z/OS DETACH of its subtasks.
In these situations a S33E ABEND can occur. This often is more noticeable in situations where the subtasks are communicating with external address spaces from the CICS address space to cause a "file Close" (such as to Transactional VSAM or IAMRLS), or to a "database disconnect" with IMS DBCTL, or to a DB2 subsystem, or to another third-party database subsystem.

It is recommended that the DELAY value in one of the three sample programs be increased by an extra amount of time to insure that all subtasks have sufficient time to process all closes and disconnects. Then subsequently monitor the shutdown time to reduce this value incrementally to the minimum time required to insure a clean and orderly shutdown of the CICS regions.

Using IAM/PLEX or IAM/RLS allows CICS online regions to concurrently share IAM files with other CICS regions and batch jobs with read write integrity requires minimal change to CICS. The IAM file is actually owned and opened within the respective IAM/PLEX or IAM/RLS address space. IAM provides some CICS exit routines so that IAM can recognize and properly handle key points in a unit of work and transaction processing.

To insure the integrity of the IAM file, CICS exits are activated by an IAM initialization module that is defined in the CICS PLT that is invoked during CICS initialization. The entries that are required are:

- DFHPLT TYPE=INITIAL,SUFFIX=X1
- DFHPLT TYPE=ENTRY,PROGRAM=IAMXCINI <<< req’d for CICS/TS V3.1 +
- DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
- DFHPLT TYPE=ENTRY,PROGRAM=IAMXCINI
- DFHPLT TYPE=FINAL
- END

Please note the additional entry for IAMXCINI which MUST be before the entry for DFHDELIM. This only applies for CICS/TS V3.1 and higher.

Additionally, the IAM load library that modules IAMXCINI, IAMBCICS, IAMXFCBO and IAMXFCRQ are contained within must be part of the DFHRPL concatenation for CICS. This will allow for using IAM/PLEX and IAM/RLS functionality in the CICS region. The IAMXCINI program will install and activate the IAM provided Task Related User Exit (TRUE) “IAMBCICS”, the IAM provided Global User Exits (GLUE) “IAMXFCBO” at the CICS File Control XFCBOUT exit point and IAMXFCRQ at exit points XFCREQ and XFCREQC. The four modules IAMXCINI, IAMBCICS, IAMXFCBO and IAMXFCRQ are defined to CICS in DFHCSD with the RDO attributes of:

```
LANGUAGE=ASSEMBLER
RELOAD=NO
DATALLOCATION=ANY
EXECKEY=CICS.
```

Refer to Section 20 on IAM/RLS and Section 25 on IAM/PLEX for additional information on using CICS with the IAM record sharing services.
10.50 IAM Tuning Guidelines

Overview

For the great majority of datasets converted to IAM, there is no real need to perform any tuning. This is because IAM, with its Real Time Tuning capability and Turbo buffering mode, can generally provide an outstanding level of performance, without the need for any manual intervention. The tuning guide is being provided for those installations that want to make sure that they are getting the best possible performance from IAM. It is also being written to address those few files that just seem to require a bit of extra effort. The last reason is to provide an aid for installations that are having a resource constraint, particularly with real or virtual storage.

What do we mean by performance? With an access method, such as IAM, performance is retrieving and storing data faster while using less computing system resources. Nothing is free however, and frequently there are trade-offs involved, such as using more of one resource to use less of another. These resources include the utilization of the processor, referred to as CPU time, DASD space utilization, and utilization of channels, control units, and physical devices to move the data between processor storage and the device, and the use of both virtual and real storage. An access method needs to use a portion of all of these resources to provide the service. Frequently tuning involves adjusting resource use from a constrained resource to an unconstrained resource.

The longest portion of any I/O operation is the time it takes to transfer data from the storage device into processor storage. It therefore follows that the fastest logical I/O is one where there is no physical I/O. IAM’s goal is to satisfy as many logical I/O requests as possible without performing any physical I/O. IAM generally utilizes virtual storage, and of course the underlying real storage, to obtain the high performance. The two primary storage areas to accomplish that are the index area and the buffers. IAM retains the entire index, normally in an internally compressed format, for the dataset in virtual storage while the dataset is open. This eliminates any need for physical I/O to read the index. Any record can be retrieved with no more than 1 physical I/O. With VSAM, assuming the required control intervals are not within the buffer pool, a typical random read will generally require at least 2 physical I/O’s, one for the index and the other for the data. Depending on the VSAM index structure, more I/O’s may be required for the index component.

The other way IAM reduces physical I/O’s is through its proven Real Time Tuning buffer management algorithms. Using dynamic buffer management techniques, IAM is able to reduce physical I/O to a level that is quite difficult to achieve with VSAM, even with extensive manual tuning. IAM will dynamically adjust the number of buffers, the buffer management algorithms, and physical I/O techniques used to match the application programs current requirements. The result with IAM is better performance, resulting in reduced elapsed times for batch jobs, reduced response times for online transactions, and less time investment required on tuning to achieve those goals.
Global Option Settings

The easiest way to minimize the need for manual tuning is to use a good base of default parameters. With IAM, these parameters are provided through the IAM Global Options Table. The default Global Option values selected for IAM Version 9.2 will provide a high level of performance for the great majority of IAM datasets and users. Provided below is a discussion of considerations for various Global Options that have a direct impact on IAM performance.

IAM defaults to using data compression for any dataset that is defined as being 75 tracks (5 cylinders on a 3390 device) or larger. The benefits are reduced DASD space requirements, reduced virtual storage requirements for indexing the dataset, and reductions in physical I/O because there is effectively more data in each block. The cost is additional CPU time to process the dataset. For many datasets with record sizes of around 500 bytes or less, IAM CPU time with data compression is still lower than VSAM without compression.

When the key performance objective is to reduce CPU time, then either limiting or eliminating the use of data compression is a good choice. Data compression can be effectively turned off for all except ESDS type of files by setting the Global Option DATACOMPRESS=99999999. Some users have limited data compression to only the largest files where space savings are significant by raising the default value. For example, only allowing files that are 500 cylinders or larger can be done by setting the Global Option DATACOMPRESS=7500.

The decision as to whether or not to use data compression is of higher relevance than the actual type of compression (software or hardware). Regardless of the type of compression used, there is going to be a CPU time cost. The key performance difference is that hardware compression has higher CPU time overhead on file loads and reorganizations, but slightly lower CPU time than software on file access. From a performance perspective, if the IAM files are subject to frequent reorganizations or file loading then software compression is the better choice for CPU time considerations. For files where loading and reorganizations are infrequent then hardware compression is the better choice for CPU time considerations. Space savings is going to vary based on the data, for some files software will do better, and for other files hardware will do better. A customized hardware compression dictionary built with CSRBDICT (see Section 10.25) will generally provide the most space savings. The default compression type is specified by the COMPRESSTYPE Global Option.

The basic recommendation for data compression is to decide on your performance objectives, and set the DATACOMPRESS Global Option appropriately. Many users go with the defaults to gain some DASD space savings. If CPU time reduction is of high importance, as it is for many installations, then globally turn off data compression by setting DATACOMPRESS=99999999.
Buffer Space

The values for BUFSP and CICSBUFSP as IAM is shipped are the same as IAM Version 9.1. Please note that the values are specified in kilobyte units (1024 bytes) as opposed to at the byte level. The default for batch and TSO buffering is BUFSP=65536. (64 megabytes)IAM calculates the default MAXBUFNO from this Global Options Value. This recommended value causes MAXBUFNO to go up to just over eighty cylinders worth of buffers on 3390 types of devices. IAM will start out with approximately one cylinder’s worth of buffers per file however it will not exceed the size of the dataset. Subsequent to OPEN, with IAM’s Turbo mode set (which is the default), IAM can quickly increase the number of buffers if necessary for files that have very heavy I/O requirements. Users desiring to be less aggressive or that have batch jobs with more than 30 IAM files that are heavily accessed may want to reduce this value to perhaps 32768.

For CICS regions, the default is CICSBUFSP=1024 (1 megabyte). This value will be used by IAM to calculate MAXBUFNO when running under CICS. This Global Option was added so that customers could give their batch work a larger quantity of buffers automatically, while not giving as many to CICS due to the frequent virtual storage constraints that are encountered under CICS.

The tuning recommendation for CICS is to use a CICSBUFSP value that provides good performance for the majority of files, and then utilize the IAM overrides to increase buffers for the files that are subject to very high activity. This strategy helps to control storage use which is quite critical for CICS regions.

With IAM Version 9.0 some changes have been made for buffer management during open processing. For large files it is beneficial to have a lot of buffers during open to facilitate reading the index into storage. IAM will reduce the number of buffers at completion of file open processing to either MINBUFNO if it has been overridden or to the number of strings (STRNO) that were specified for this dataset. This will prevent the circumstance for files that have very low activity from having a lot of buffers that are not being used and wasting virtual storage. With this change, more users may be able to leave CICSBUFSP at the default value rather than reducing it, and minimize the need for overrides.

The default values for BUFSP and CICSBUFSP were chosen, along with defaulting to Turbo mode, to reduce the need for using overrides to increase buffering. Installations can adjust these values as necessary to achieve their desired performance results.

64-Bit Virtual Storage for Buffers

New in Version 9.2 is the ability to utilize 64-bit virtual storage for buffers. This is ideal for use with CICS, IAM/RLS or IAM/PLEX regions that have a lot of open IAM files and are experiencing some 31-bit virtual storage constraints. It can be set as a default by the IAM Global Options, however as it does cause some increase in CPU time Innovation recommends that it be used selectively when needed.

Below the Line Storage

Below the line storage utilization is also a critical resource for CICS. IAM typically utilizes 4K of below the line storage per open file. This is significantly reduced through the use of the BELOWPOOL feature, which allows IAM to share the I/O control blocks and channel programs between multiple files, reducing this storage requirement. This is controlled with the BELOWPOOL Global Option. With IAM Version 9.0 this function has now been enabled as a default, which is BELOWPOOL=YES. This default is highly recommended.
File Load Buffering

The next recommendation is to leave CRBUFOPT=MCYL, to cause a file load to buffer up to one cylinder worth of data, while physically writing a full cylinder of data. This is the optimal buffer setting for file loads. CICS users should consider specifying a CREATE override such as the following:

```
CREATE DD=&ALLDD, CRBUFOPT=TRK
```

This will help reduce storage in case empty files are opened under CICS.

ESDS Processing

If your installation is using IAM ESDS types of datasets that are updated, then be sure to set ESDSINTEGRATED=5. This will allow some room for record updates that require more space after data compression without having to use the Extended Overflow area. The cost is more DASD space usage to load the dataset initially, but if the dataset is updated, this will prevent the use of DASD space, virtual storage, and I/O for the Extended Overflow area.

Variable Overflow

The recommendation is to leave this Global Option set to the default value of VAROVERFLOW=YES. This enables IAM’s variable overflow that will result in more data records in each overflow block. This reduces DASD space requirements for the overflow area, plus may also help reduce physical I/O’s to the overflow area. The savings, particularly for datasets defined with very large maximum record lengths can be substantial. The disadvantage is that records that are repeatedly updated may have to be moved to a different overflow block from time to time if the record length increases.
General Considerations for Optimum Performance

Most of the IAM datasets will achieve outstanding performance results, particularly with the recommended Global Options settings. In general the IAM files do not require any routine monitoring. If performance is not up to your expectations, or if you want to make sure that you are receiving the best performance possible with all of your datasets then consider the following general guidelines.

1. Make sure that you have a way to obtain IAMINFO reports. This can be done either by adding an IAMINFO DD card to the JCL for job steps using IAM files or by collecting the IAM SMF records, then post processing the data with the IAMINFO command of the IAMSMAF utility program. Use of the IAMSMAF program is recommended particularly because it can generate a CSV format file of the IAMINFO report data to the data can be analyzed in spread sheet format. These reports contain critical information for detailed tuning. Become familiar with the contents of these reports, as they provide a lot of useful information. By activating and collecting the IAM SMF records, you can also utilize the IAMSMAFVS reports for a more concise report format that will make it easy to find those datasets that might require additional attention.

2. Periodically review the IAMINFO reports. If more buffers would have helped reduce physical I/O’s, the IAMINFO report will contain an IAM368 message indicating so. In fact, you can request that IAMSMAF print off only those IAMINFO reports where that message appears, with the keyword ATTRIBUTE=MOREBUFFER. If this message is appearing for several datasets, then perhaps the BUFSP Global Option should be increased to avoid the need for several overrides.

3. For optimal I/O performance, use data compression. This will help reduce physical I/O’s, reduce virtual storage for the prime index, and reduce DASD space requirements. However, if CPU time is more of a concern, then don’t use data compression particularly on datasets with long records, or that are sequentially processed.

4. Make sure that heavily updated files are regularly reorganized. This will help prevent virtual storage problems, long open times, and high physical I/O activity.

5. Avoid the specification of Share Options 3 or 4 for IAM datasets, unless you have activated IAM/PLEX or IAM/RLS. The optional IAM/PLEX feature provides support for sharing files for update across multiple LPARs in a SYSPLEX, and IAM/RLS does the same for applications running within the same LPAR. Specification of those share options without IAM/PLEX or IAM/RLS will force additional physical I/O’s that can be substantial and will not maintain full data integrity.

6. Investigate increasing the block size for datasets with a large Prime Index structure, particularly if the dataset has relatively large record sizes.

Buffering

IAM makes it easy to determine when more buffers could have reduced I/O by providing the IAM368 message in the IAMINFO report. Unless there is a concern about storage, there is no reason to be concerned about being overly aggressive at setting MAXBUFNO. IAM’s Real Time Tuning will carefully adjust the buffering for the dataset as processing needs vary. For programs that do all sequential processing the maximum number of buffers used for the file will be the number of blocks per cylinder plus a few additional buffers to handle Extended Overflow blocks. Usually providing one or two tracks worth of buffers for overflow will be sufficient, unless a dataset has a very extensive use of overflow. Using the default BUFSP Global Option setting will handle setting the defaults to maximize the buffering for sequential processing, eliminating the need to increase buffers for most batch jobs.
For programs that do all random I/O, a mix of random and sequential I/O, or short strings of sequential I/O requests, then the MAXBUFNO value should be increased by a quantity that you feel comfortable with. If you are not concerned about virtual storage usage or paging, then use a large quantity. If however storage is of a concern, increase the value by 4 or 8, and see how that helps. The methods of increasing MAXBUFNO for any particular file include:

- Providing an IAM ACCESS MAXBUFNO or BUFSPACE override for the job step and dataset.
- Specify the BUFND parameter, either within the ACB or as part of the AMP parameter on the DD card for the dataset, e.g. AMP=('BUFND=nnn'). For CICS files not in a LSR pool, the resource definition for the data buffers will result in changing the BUFND value in the ACB that is used by CICS.
- To increase buffering for all jobs that use this dataset, provide an IAM CREATE MAXBUFNO override when the dataset is defined, loaded, or reorganized. The specified MAXBUFNO value will be applied whenever the dataset is accessed.
- Specify a value for BUFSPACE on the IDCAMS define control statement for the dataset.

The two circumstances where you might not want to increase buffers for the job are:

1. When the job is performing sequential processing against a dataset that is concurrently open to online systems. This is because the batch job could end up dominating the file, resulting in poor response times for users of the dataset on the online system. In fact, you will probably want to reduce MAXBUFNO for such jobs.
2. When the job has a virtual storage constraint. Refer to the Section 10.51 on Storage Usage for adjusting buffers with jobs that have storage constraints.

Otherwise, it is perfectly fine to increase the MAXBUFNO or BUFSPACE value.
Excessively large Extended Overflow usage can result in a deterioration of performance that can usually be avoided. These problems are avoided by the periodic reorganization of files when they are using some large quantity of extended overflow. VSAM datasets also require reorganizations due to performance deterioration and space usage. Because of that, many application job streams that were converted to IAM from VSAM already have regularly scheduled file reorganizations, which will generally be sufficient for IAM datasets. Depending on the dataset and application activity, the reorganizations may be done daily, weekly, monthly, or even quarterly.

Some of the symptoms of an excessively large usage of extended overflow are:

- Long elapsed time to open the dataset.
- Excessive use of virtual storage or the IAM Index Space.
- High I/O rates when processing the dataset sequentially.
- Potential inability to open the dataset or other datasets due to virtual storage constraints.

It can be difficult to predict the level of extended overflow usage at which serious performance deterioration will occur. For example, one file could have well over a million records in overflow, and not be experiencing any noticeable performance difficulties, whereas another dataset may only have a few hundred thousand records in overflow and be experiencing severe symptoms. The key factors are the key length and the general placement of records in the overflow area. For example, if a file has a key length of 4 with a million records in overflow, the storage used for that index is going to be substantially less than if the file had a key length of 64. If the records in overflow are in a generally ascending key sequence, or in clusters of ascending key sequence, then the I/O impact and processing time to open the dataset will most likely not be seriously impacted. A very random pattern of records throughout overflow can have a serious impact on sequential I/O performance, and the processing time to open a dataset.

If there is a need for frequent reorganizations, then consider using such files with the Prime Related Overflow (PRO) format. The PRO format has a different structure to the overflow area that is indexed at the block level rather than the record level, generally resulting in a much smaller index and reduced frequency to reorganize the file. The use of PRO is recommended for files that have around a million or more records in overflow.

One of the cautions is for reorganizations that are done by application programs. Some application programs reorganizations are done by a single record load followed by a mass insert. This is not a valid reorganization from the access method point of view. The resulting dataset will frequently be in a less than optimum status after the application reorganization. If such a technique is used, the application reorganization should be followed by a file reorganization that is done by FDRREORG or an IDCAMS REPRO.

The other thing to watch out for on these scheduled reorganizations is where within the batch job stream they occur. For example, some applications reorganize a dataset after they are closed online, and then execute a large batch update process. The batch update process can result in heavy overflow use, so that when the dataset is subsequently opened for online processing, it is in a less than optimum state. By simply scheduling the reorganization after the update processing, the file will be in the best possible organizational state when it is opened for online processing.
Guidelines for Reorganizing

Some guidelines for determining when an IAM dataset should be reorganized include the following:

- When more than 10 to 20 percent of the records in the dataset are in extended overflow.
- When the size of the Extended Overflow Index exceeds some storage quantity, such as 64 megabytes.
- When the Overflow area exceeds a quantity of DASD space, such as 1,000 cylinders.
- When a single volume dataset is approaching sixteen extents.
- When the number of overflow records for a particular dataset approaches or exceeds a predetermined number of records. IAM can assist in monitoring this if the file is defined with an Overflow (O=) override of that specified value.

For some of the above guidelines, IAM will issue informational messages. These messages have been changed in Version 9.2, and are now the IAMW21, IAMW22, IAMW91 and IAMW92 indicating that reorganization is recommended along with the reason. The messages also use the z/OS multi-line format to easier identify the dataset name with the message. This will make it easier for users that schedule reorgs based on the IAM messages to be more selective in the files that they trigger the reorgs for. The IAMINFO reports will also include an IAM373 message indicating that reorganization is recommended. Several of these factors are available as selection criteria on FDRREORG, which provides an automatic method for reorganizing files only when needed. Other methods of automating file reorganizations include using the reports generated from the IAM SMF records, by IAMSMF or IAMSMFVS. Full information on IAM dataset reorganizations is provided in Section 10.81 of the IAM Manual.

Large Prime Index

Datasets that have a prime index structure that exceed a few megabytes are considered to have a large prime index. The amount of storage required for the Prime Index, and whether or not it is compressed, is provided in both the IAMINFO reports and the IAMPRINT reports. Having a large prime index structure will not necessarily cause a performance problem, however such files may realize improved performance by reducing the prime index size. There are a number of factors to consider. The potential advantages of reducing the prime index size include faster index search time and reduced virtual storage requirements. The reduction in virtual storage may be partially, or in some cases entirely, offset by an increase in buffer size if the block size is increased. The costs are increased search time for records within each data block and increased physical I/O time.

The prime index size is based on the number of prime blocks, the key length, and the compressibility of the key structure. From a tuning perspective, there are two possibilities for reducing the index size: improving the index compression and/or reducing the number of prime data blocks.
Controlling Index Compression

There are some parameters that influence the key compression for the index that are in the IAM Global Options Table, or also available as IAM Overrides. Basically, for any particular data set, IAM will take a fixed number of the high keys in each block, and compress them by elimination of each relative position in the keys that have an identical value. For example, if the first byte of each key is A, then A will be eliminated for that set. If the second byte of each key is B, then the second byte is eliminated from each key in the set. This process continues for each byte of the key, leaving only those byte positions where all the keys in the set do not have the same value. The number of keys in each set when the key length is less than 128 bytes is the lower of the number of keys that will fit in 256 bytes, or the value for LIMITKEYS. The default value for LIMITKEYS is 32.

For data sets with fairly long keys, there is the Long Key Compression option. Long Key Compression eliminates the limitation that a set of keys must fit in 256 bytes, and may provide improved index compression for files with long keys. The default minimum key length for Long Key Compression is 33 bytes. Regardless of the key size, if it is compressed with Long Key Compression then the number of keys in each compressed set is completely controlled by the LIMITKEYS value.

When using the Long Key Compression the value for LIMITKEYS can be adjusted on a file by file basis using the LIMITKEYS override. Some experimentation may be necessary to see if using a lower or higher value will help to improve the amount of index compression that can be achieved.

One can also change the Global Option value for when Long Key Compression is used, such as reducing it down to 9.

Reducing the Number of Prime Blocks

You may have some control over the number of prime blocks. The first thing to make sure of is that the data compression is enabled for the dataset. This can help reduce the number of prime blocks by fitting more data within each prime data block. The next factor to check is for a large CI free space area. Make sure that such a large CI free space area is warranted based on insert or record growth activity to avoid overflow growth. For most files providing CI free space results in wasted space. Reducing a large CI free space will result in fewer prime blocks. Next, if the file is at less than 1/2 track blocking, increasing the block size will reduce the prime index storage. Changing the block size requires some caution, unless the dataset is quite predominately sequentially processed. Random processing or short sequential browses that are typical of online systems may incur increased response times when using a larger block size, due to the increase in data transfer time. Plus they are also subject to increased CPU time to search the data block for the required record.

When to Increase the Block Size

So, when is it beneficial to increase the block size? There are two factors to consider. The first is the average record size as the data is stored, and the second is the benefit of buffering. As record sizes increase, there will be more benefit to increasing the block size providing that buffering is reducing physical I/O. The average stored record size is provided on the IAMINFO report for the file load. If that is not readily available, then the approximate value can be calculated with data from an IAMINFO or IAMPRINT report as follows:
The benefit of buffering can be easily determined from data in the IAMINFO report by dividing the Disk Blocks Read by Requests Processed. This presumes of course that an adequate number of buffers are being provided. As this percentage of requests requiring I/O gets smaller, the benefit of buffering is increasing. The larger the benefit of buffering, the more likely it is to achieve benefit by increasing the block size. There may not be much benefit, from the physical I/O perspective if more than 50% of the logical requests require I/O. The I/O benefit is likely to be larger as the percentage drops to 25%, 10%, or even lower.

How to Increase the Block Size

As a general rule, if the average record size is 1K or more (1024 bytes), and there is some beneficial buffering, there should be no hesitancy about increasing the block size. The block size, or blocking factor, can be changed by either using the CREATE IAM Override B= during the file define, load, or reorganization to specify a block factor. The alternative is to increase the CI size on the Define statements. For example, specifying a B=2 override will force half-track blocking. A blocking factor of 1 is not recommended because there will be a considerable amount of DASD space wasted due to the limitation of the IAM dataset block size of 32K.

```
//IAMOVRID DD *
CREATE DD=&ALLDD,B=2
/*
```

For datasets with smaller average record sizes, increasing the block size can be considered and will be beneficial with larger prime index structures as long as there has been beneficial buffering. There probably is not much benefit to increase the block size for files with average record sizes of less than 500 bytes, unless the I/O activity is predominately sequential.
High I/O Rates

This section will discuss some of the common causes of higher than expected physical I/O's, commonly referred to as EXCP count. The IAMINFO report is a necessity to understand such a problem. The key statistical fields from the IAMINFO report that are used include the following:

- **DISK BLOCKS READ**: The number of physical I/O's (EXCP's) that were issued to read data from the IAM dataset.
- **DISK BLOCKS WRITTEN**: The number of physical I/O's (EXCP's) that were issued to write data to the IAM dataset.
- **SEQ CHAINED BLOCKS READ**: The number of additional data blocks read in as part of a sequential I/O. This number plus the **DISK BLOCKS READ** is the total number of blocks read into storage.
- **SEQ CHAINED BLOCKS WRITTEN**: The number of additional data blocks written out as part of a sequential I/O. This number plus the **DISK BLOCKS WRITTEN** is the total number of blocks written out to DASD.

The total EXCP count for the IAM dataset can be easily calculated by adding the **DISK BLOCKS READ** and **DISK BLOCKS WRITTEN** values. It is quite useful to have the two separate values, as they will help in our search for what is going on with the dataset. Some of the circumstances and potential actions are described below.

If the value for Disk Blocks Written is very high, then most likely what is happening is IAM is not deferring the writes for random updates. This situation occurs when the dataset is defined with Share Option 3 or when a Share Option 1 or 2 dataset is processed asynchronously, as is done by CICS. For online systems, this generally is a desired action so no change is recommended. For datasets defined with Share Option 3, they can be redefined with Share Option 2 because of the very high risk associated with sharing an IAM dataset for update.

If both the Disk Blocks Written and Disk Blocks Read are very high, such that they equal or exceed the total requests, the most likely cause is that the file is defined with Share Option 4, unless IAM/PLEX or IAM/RLS is active for this dataset. Setting Share Option 4 forces IAM to use only 1 buffer, and IAM will always reread a data block whenever it is requested, even if it is already in the buffer. Plus, IAM will always immediately write out any updated data block, including sequentially updated data blocks. The dataset should be redefined with a Share Option of 2, because sharing an IAM dataset for update is most likely going to result in a corrupted dataset, and lost data.

If both Disk Blocks Read and Seq Chained Blocks Read are exceedingly high, the problem is most likely that IAM is rereading empty prime or PE blocks. This can result due to an application having mass deleted a large group of records that occupied contiguous blocks, followed by attempts to retrieve records using a key greater or equal type of search. Depending on the Share Options and how the dataset was opened, IAM is able to avoid this type of processing. The affected dataset should be reorganized to resolve the problem. Try using the REREADEMPY=NO IAM ACCESS override which may prevent the high I/O rate.

If Disk Blocks Read is quite high for a basic sequential I/O type of job, then the most likely cause is that there are a lot of records in key sequence that are scattered through many different Extended Overflow blocks. Such a situation is also likely to be coupled with a long time to OPEN the dataset, due to the Extended Overflow index build process. The solution to this problem is to reorganize the dataset.
For datasets that have an unusually high I/O activity and are not on a device with PAV (Parallel Access Volume), it may be quite beneficial to spread the dataset across multiple volumes. By so doing with Enhanced Format IAM datasets, there can be concurrent physical I/O scheduled to each DASD volume, which may result in significantly improved online response times. With a little bit of planning, this is easy to accomplish by setting up proper space allocation parameters. Two different techniques for accomplishing this will be shown. For both examples, it has been determined that the dataset requires approximately 2,000 cylinders of space, excluding overflow requirements. The bulk of the dataset will be split across 4 DASD volumes, however a fifth volume will be used to handle any potential growth into the IAM Extended areas.

The first example can be used for installations that have DFSMS active on their system. Note that the dataset does not have to be SMS managed for this technique to work, just have to have DFSMS active. If the dataset is going to be on SMS managed volumes, then the dataset must be defined with Guaranteed Space. If the dataset is being allocated to non-SMS managed volumes, then IAM allocates the dataset as if it were being defined with Guaranteed Space under DFSMS. That is, IAM will allocate the primary space quantity on each volume when the dataset is defined. For this technique to work, the secondary space quantity must be 0, which will prevent the usage of secondary extents. File expansion is accommodated by utilization of the space on the fifth volume.

```
//DEFCMD EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=* 
//SYSIN     DD   *
   DEFINE CLUSTER   -
      (NAME(MY.IAM.KSD)   -
      OWNER($IAM)   -
      VOLUMES(MYVOL1 MYVOL2 MYVOL3 MYVOL4 MYVOL5) -
      CYL(500) RECORDSIZE(100 1000) -
      KEYS(24 8) FREESPACE(5 20) -
      SHAREOPTIONS(2 3) REUSE )
   LISTCAT ENT(MY.IAM.KSD) ALL
/*
```

**Figure 10-35: Example of Spreading IAM Dataset across Multiple Volumes**

In the next example, a different technique is used in that the dataset will be allowed to take secondary extents. This is effective for files that are not SMS Extended Format files. To achieve the desired split of 500 cylinders across 4 volumes, a primary of 200 cylinders is being requested, with a secondary of 20 cylinders. The secondary results in a total of 300 cylinders, being 15 extents times 20 cylinders. The IAM overrides of MAXSECONDARY=1 is specified to prevent IAM from increasing the secondary allocation, and an override of MULTIVOLUME=PRIMARY is specified to cause IAM to allocate the primary on the next volume for the first extent.
//DEFMULTV EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//IAMOVRID DD *  
CREATE DD=&ALLDD,MAXSECONDARY=1,MULTIVOLUME=PRIMARY /*  
//SYSIN DD *  
DEFINE CLUSTER -  
(NAME(MY.IAM.KSD) -  
OWNER($IAM) -  
VOLUMES(MYVOL1 MYVOL2 MYVOL3 MYVOL4 MYVOL5) -  
CYL(200 20) RECORDSIZE(100 1000) -  
KEYS(24 8) FREESPACE(5 20) -  
SHAREOPTIONS(2 3) REUSE )  
LISTCAT ENT(MY.IAM.KSD) ALL /*

Figure 10-36: Example of Spreading IAM Dataset Across Volumes
IAM Storage Usage

IAM Storage Usage
IAM does utilize virtual storage to provide the services requested by application programs. IAM uses virtual storage to reduce I/O and the CPU time required to process an indexed dataset. One of the features of IAM is that it keeps the index to the file in virtual storage. There are no index buffers, or index I/O after the file has been opened. Data buffers are the other major component of virtual storage usage. These large storage areas are always requested from the 31-bit addressable area of memory (above the line) or optionally from 64-bit addressable storage. This use of storage seldom is a problem for batch jobs. However, large online regions that have hundreds of files open at any point in time may run into virtual storage constraints. This section will attempt to explain IAM's storage usage. Users are encouraged to contact Innovation for IAM technical support for assistance with any storage usage issues.

Virtual Storage Management Features
To help minimize the need for virtual and real storage tuning, IAM has several special features to aid in the dynamic management of virtual storage for jobs using Enhanced Format IAM datasets. IAM can put the index for an open IAM dataset into either 64-bit virtual storage or a z/OS data space, which is referred to as an Index Space. This feature alleviates virtual storage contention by moving a lot of IAM storage into storage areas without the constraints of normal address space storage. IAM can also dynamically increase the above the line storage region based on values from the IAM Global Options Table. This provides for a quick way to dynamically adjust to unexpected storage requirements without having to change the IEFUSI exit or the job's REGION parameter. When acquiring a non-critical area of storage, such as acquiring an additional buffer, if the storage was acquired below the 16-megabyte line, IAM will release that storage. This way, once the above the line region is filled, IAM will not unnecessarily use below the line storage, which could quickly disappear. IAM also monitors buffer usage, as a part of the Real Time Tuning, and will release infrequently referenced buffers as long as there is some I/O activity on a dataset.

New in Version 9.2 is the ability to use 64-bit addressable virtual storage for the I/O buffers. This can be done either by using the ACCESS override of BUF64=YES and the CREATE override of CRBUFOPT=64BIT. The IAM Global Options could also be changed to make this the default values. Be aware that using the 64-bit virtual storage for buffers can increase the amount of CPU time being used, so if that is of concern use this option only when necessary. The MELIMIT value may also need to be increased when using the 64-bit virtual storage for buffers as well.

Base Storage Requirements
The actual amount of storage used to process each file, excluding load modules is provided on the IAMINFO report. This includes the amount of the total storage required and the amount of that storage which was above the line. IAM will generally allocate virtual storage in multiples of 4K size areas, and manage the allocation of that storage. This is done to help prevent storage fragmentation and to improve reliability by reducing the chances of storage corruption that can easily occur when multiple programs are sharing the same page of virtual storage.

The minimum storage requirement per open Enhanced format file is 24K, which is divided into five separate areas, not including the index, buffers, and load module storage. The amount of above the line will vary quite a bit depending on the options being used. The amount is larger than in prior releases due to support for z/HPF, format 1 CCW's, the IOBE and IEDB which were not used in prior versions and 64-bit virtual for buffers.
IAM Storage Usage

**Below the Line Storage**
IAM limits usage of 24-bit addressable memory (below the line) as much as possible. IAM generally requires only 4K of storage below the line to handle the I/O control blocks. In version 9.2 the actual channel programs have been moved above the line due to use of format 1 CCW’s. The IOB’s and DCB still need to be below the line. This amount may be larger if more IOB’s are needed, or if the file has a very large number of extents. Note that the initial number of IOB’s obtained is based on the STRNO value provided. For CICS with the BELOWPOOL=YES Global Option the amount per file will be considerably less than the 4K.

IAM requires approximately 4K of virtual storage to hold the simulated VSAM control block structure, which may reside either above or below the line, depending on what was specified in the ACB. The base VSAM control block area is 2352 bytes for a KSDS type of file, or 752 bytes for an ESDS plus the storage required for each string, or place holder. This area can also exceed 4K, if a larger value is specified for STRNO, which indicates the number of place holders. The place holder size is also impacted by the key length, as described above. For CICS, the VSAM control block area is above the line in 31-bit addressable storage.

**CICS Below the Line Storage Reduction**
IAM provides a capability to reduce IAM’s use of below the line virtual storage under CICS for opened Enhanced Format IAM files. As stated above, IAM typically uses 4K of below the line storage for each opened Enhanced Format file to contain I/O control blocks (DCB, IOB, and JFCB) and channel programs. With the Below the Line Pooling feature enabled, IAM storage usage will be reduced by at least 50% for IAM Enhanced Format files, when several files have been opened. The feature is enabled by setting the IAM Global Option BELOWPOOL to YES, which is the default value as IAM is shipped. The storage is managed similar to a z/OS cell pool type of structure. Because of this, the storage areas for the pools are retained as IAM files are closed, but are available for reuse by IAM processing when IAM files are reopened.

**Above the Line Storage**
IAM keeps all the rest of the required control information, buffers, and index above the line. The base IAM control block area requires 4K. IAM has a buffer table that will fit within an additional 8K as long as the MAXBUFNO value does not exceed 128. That table has increased in size due to the need to keep 64-bit addresses. There is the prime block to overflow table, which has a minimum size of 4K, and may be larger depending on the file size. There are some work areas for data compression, index decompression, and the high level extended index which depending on their size requirements may fit within the other IAM storage areas.

**Buffer Storage**
The buffer storage is broken down into single block areas, the size of each is the block size rounded up to a 4K value. A buffer for a typical ¼ track blocked IAM dataset on a 3390 type of device requires 16K. The maximum buffer storage used is easily calculated by taking the buffer size value, and multiplying it by the maximum number of buffers used from an IAMINFO report. Whenever IAM is acquiring a buffer, if the storage assigned is below the 16-megabyte line, IAM will release the storage.

When the buffer storage is in 64-bit addressable virtual storage, the storage is acquired in 1 megabyte increments. Because it is managed in larger increment than 4k page, the number of buffers contained in a megabyte is somewhat higher. For example a typical 1/4 track blocked file will have 76 buffers per megabyte in 64-bit virtual storage, whereas due to rounding each buffer to a 4k page size, 64 buffers are in a megabyte of 31-bit addressable storage. IAM manages storage in page size boundaries to reduce storage fragmentation issues in 31-bit virtual storage.
Index Storage

There are three different index areas for an IAM dataset. The first is the prime index, which is created when the file was loaded. The second is the index for the Extended PE area, and the third is for the Extended Overflow area. Both the second and third index areas are dynamic, and will change as the file is updated. The Extended PE index is saved within the dataset, but the Extended Overflow index for the basic file format is always dynamically built during open processing. The Extended Overflow index for PRO format files is saved in the dataset and read in when during open processing.

Prime Index

The prime area index is fixed in size at completion of the file load or reorganization, and will never be updated. This index is based on the high key in each prime block. IAM provides the capability to compress the prime index in a proprietary form that can greatly reduce the amount of storage required for the index. Index compression is an automatic feature that will be used whenever the prime index exceeds 8000 bytes, and the attributes of the key fit within the compression criteria. The amount of storage required for the prime index is provided in the IAMINFO report and on the LISTCAT IAMPRINT report. Take the indicated value from one of those reports, and round it up to 4K to determine the amount of storage that will be used for the prime index. Refer to Section 10.50 for additional information on the prime index, and controlling the prime index size.

Extended PE Index

The next index storage area is for the Extended PE blocks. This index, like the Prime Index, is based on utilizing the high key within each data block. The size of each entry is the key length plus four bytes, with an entry for each Extended PE block. Due to the internal structure of the Extended PE index, which is organized based on an internal grouping of the extended index blocks, the total storage used for the Extended PE is difficult to predict. Files with large quantities of Extended PE blocks, which are clustered together may not necessarily use any more storage than a file with a few Extended PE blocks that are sporadically spaced throughout the extended area of the file. While this index is not compressed, it is still a relatively efficient format especially because only a very few files actually have need for this index.

Extended Overflow Index

The last segment of the index storage is for the Extended Overflow blocks. This is a record based index structure, consisting of an entry for each record in overflow. This index is subset into smaller groups, where each grouping consists of the overflow records from a particular prime block. An overflow index search is only done once it has been determined that the prime (or Extended PE) block that should contain the record has associated overflow records. This type of structure is expected to reduce the number of overflow index searches, reduce the number of entries any single search has to scan, and reduce the IAM CPU time for many functions related with overflow.

Estimating the actual storage requirements for the overflow index is difficult. The entries within each subset have compressed key format, but each subset also has header information. Each subset may have some empty entries. As a rough estimate, add four to the key length, and multiply that by the number of records in overflow. The result is the size of the overflow index prior to compressing, which may reduce the storage requirement, although the headers for the subsets will increase the storage requirement.

PRO Overflow Index

When using the PRO overflow format, the index is based on one key per overflow block. This structure may significantly reduce the amount of storage required for the overflow index however it comes with a cost of using more disk space than the standard Extended Overflow. PRO format is recommended for files with a large number of overflow records, approaching a million or more.
The IAM Index Space virtual storage can reside in either 64-bit addressable virtual storage in the user address space, or as a z/OS Data Space. IAM will place all of the index areas for a file in one of those Index space areas if available. By relocating these potentially large index areas into the IAM Index Space, there is more virtual storage available within the 31-bit and 24-bit addressable areas of the job step region. This is expected to be of benefit to large online regions, which may have several large IAM files open. The IAM Index Space storage is dynamically acquired by IAM when the first file is opened that will be using the Index Space, and is retained until the job step terminates. For any job step, there will be only one IAM Index Space, either in 64-bit virtual or a data space, with all open IAM files using storage from the same Index Space. The advantage of 64-bit virtual storage is the elimination of the maximum 2 gigabyte size of a data space which was becoming a constraint for some users.

As IAM is shipped it will by default use 64-bit virtual storage if available. If insufficient 64-bit virtual storage is available, then IAM will try to acquire a data space. If data space storage is not available, then IAM will use 31-bit and potentially 24-bit virtual storage for the index. This can be changed by either changing the IAM Global Options, or on a job step and file by file basis with the IAM ACCESS Override INDEXSPACE.

When using a z/OS data space the size of the data space requested is taken from the IAM Global Options Table value for DATASPACE. The Index Space is created to be extendable, with the maximum size set to the smaller of four times the DATASPACE value or 2048 megabytes. The default value for the data space size is 2048 megabytes. Specification of sizes less than 512 are not recommended because it may cause failures that could have been avoided.
Shown below is a list of the modules required for accessing Enhanced Format IAM datasets, with their approximate virtual storage requirements. Only one copy of each module is loaded, as required, regardless of the number of IAM datasets opened by a task.

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Storage Required</th>
<th>RMODE</th>
<th>LPA Eligible</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAMABUFR</td>
<td>36K</td>
<td>ANY</td>
<td>YES</td>
<td>IAM buffer manager and physical I/O driver.</td>
</tr>
<tr>
<td>IAMACCIX</td>
<td>15K</td>
<td>ANY</td>
<td>YES</td>
<td>IAM AIX and Path Logical I/O Request Handler</td>
</tr>
<tr>
<td>IAMACCKS</td>
<td>108K</td>
<td>ANY</td>
<td>YES</td>
<td>IAM Logical I/O Request Handler.</td>
</tr>
<tr>
<td>IAMACCXM</td>
<td>17K</td>
<td>ANY</td>
<td>YES</td>
<td>IAM Logical I/O Request Handler for IAM RLS</td>
</tr>
<tr>
<td>IAMADNAC</td>
<td>1K</td>
<td>ANY</td>
<td>NO</td>
<td>IAM Anchor.</td>
</tr>
<tr>
<td>IAMAHPF1</td>
<td>4k</td>
<td>ANY</td>
<td>YES</td>
<td>Builds IAM z/HPF channel programs</td>
</tr>
<tr>
<td>IAMAPTOC</td>
<td>30K</td>
<td>ANY</td>
<td>YES</td>
<td>IAM Path and AIX Open and Close.</td>
</tr>
<tr>
<td>IAMAVSOC</td>
<td>76K</td>
<td>ANY</td>
<td>YES</td>
<td>IAM Open, Close and support subroutines.</td>
</tr>
<tr>
<td>IAMAVS24</td>
<td>3K 24</td>
<td>YES</td>
<td></td>
<td>IAM interface to application program and user exits.</td>
</tr>
<tr>
<td>IAMAXTND</td>
<td>15K</td>
<td>ANY</td>
<td>YES</td>
<td>IAM routine to acquire an extent. As needed.</td>
</tr>
<tr>
<td>IAMCOMPH</td>
<td>2K</td>
<td>ANY</td>
<td>YES</td>
<td>IAM hardware compression routine.</td>
</tr>
<tr>
<td>IAMCOMPO</td>
<td>7K</td>
<td>ANY</td>
<td>YES</td>
<td>IAM software Data Compression Routine.</td>
</tr>
<tr>
<td>IAMCRTVS</td>
<td>49K 24</td>
<td>YES</td>
<td></td>
<td>IAM File load processor.</td>
</tr>
<tr>
<td>IAMNINFO</td>
<td>32K</td>
<td>ANY</td>
<td>YES</td>
<td>IAMINFO Report Generator is loaded only if there is an IAMINFO DD card.</td>
</tr>
<tr>
<td>IAMOPT</td>
<td>2K 24</td>
<td>NO</td>
<td></td>
<td>IAM Global Options Table.</td>
</tr>
<tr>
<td>IAMOVRIX</td>
<td>16K 24</td>
<td>NO</td>
<td></td>
<td>IAM Override Processor used for Enhanced Format files. Will only be loaded when there is an IAMOVRID DD card and Enhanced format files are opened. Acquires a 48K table in above the line storage to hold the overrides.</td>
</tr>
</tbody>
</table>

For processing typical data compressed Enhanced format files, without any IAM overrides, there is a requirement of approximately 5K of below the line storage, and 265K above the line for the IAM load modules. If there are overrides, then the below the line storage will increase to 21K, and above the line storage to 317K. Additional storage will be necessary during file open, close, and extend processing.
Using IAM Alternate Index Support

Overview

The optional IAM Alternate Index support allows you to define and use IAM Alternate Indexes and Paths to Enhanced Format IAM KSDS or ESDS type of files. As with standard IAM support, there are no changes required to application programs and generally no changes to JCL, other than to indicate on the definition that the base cluster is to be an IAM dataset. The DEFINE ALTERNATEINDEX and DEFINE PATH can be changed similarly, but there is no requirement to do so. All the rest of the alternate index processing is essentially the same as is being done with the VSAM alternate indexes today. Some changes may be required when renaming or creating copies with different names, to synchronize all the related datasets with the new names.

Alternate Index

An alternate index provides an additional index to an indexed type of dataset (KSDS), or an index to an entry sequence dataset (ESDS). Users can define one or more alternate indexes to any base dataset. The alternate index dataset itself is an IAM KSDS enhanced format type of dataset that is indexed by the alternate key. The records in the alternate index contain some control information, the alternate key, and the primary key value for the record in the base dataset with the corresponding alternate key. An alternate index is defined as containing UNIQUE keys when there is only one base record with any specific alternate key. Alternate indexes can also be defined as containing NONUNIQUE keys, where any particular alternate key can be contained in multiple base records. As with VSAM, the alternate index dataset can be explicitly processed by programs without referencing the base cluster, or can be used to access the records in the base cluster. To use an alternate index dataset to access the records in the base cluster, a PATH must be defined.

An additional attribute of alternate index datasets is whether they are upgradeable. When an alternate index is defined with the UPGRADE attribute, IAM will automatically update the alternate index as needed whenever updates are made to the base cluster, either through the primary key or when accessed through an alternate key. Any alternate index defined with the NOUPGRADE attribute will not be automatically updated by IAM, and it is the application program's responsibility to ensure that the alternate index is updated in a manner to remain synchronized with the base dataset.

Path

A PATH provides the mechanism to access a base cluster through an alternate index. When a path is defined, one must provide the name of the path and the name of the related alternate index to be used to access the base cluster, or the base cluster to be processed whenever the path itself is referenced. The UPDATE or NOUPDATE attribute is specified for each path defined. When a path is defined with the NOUPDATE attribute, any upgradeable alternate indexes will not be updated automatically when this path is used to access a base cluster either directly or through an alternate index.

For VSAM datasets, a PATH is only a catalog entry that is quite similar to an ALIAS entry. With IAM, a PATH will become a one-track dataset containing the name of the related alternate index or base cluster and the UPDATE or NOUPDATE attribute of the PATH. A single track dataset was chosen instead of an ALIAS type of catalog entry because of concerns about there not being adequate dataset management utility program support for ALIAS entries, which might result in the loss of the entry.
Alternate Index Associations

For VSAM datasets, the associated dataset(s) in an alternate index sphere are linked together by an association cell within the catalog entry for the dataset. Because IAM datasets are cataloged as non-VSAM datasets, there are no association cells within the non-VSAM catalog entry. For this reason, IAM saves the association information within each of the associated IAM datasets. The association information consists of the dataset name and type (Base, AIX, or Path) of dataset, and includes the update or upgrade attribute. Whenever a LISTCAT is requested (or an IAM ISPF display) for an IAM dataset that is part of an alternate index sphere, the association information will be included in the output. The information in the association cell is critical, because it will direct the processing during an open of a component of the sphere, and when deleting an alternate index sphere. Note that each type of dataset includes the associated information a self-describing association entry. The associations are always listed in the order that each was defined, which will always begin with the base cluster.

Base Cluster Associations

For a base cluster, the associations will include in addition to itself, all of the alternate indexes that have been defined for that base cluster, and all of the paths that have the base cluster as its path entry dataset. Not included in the base cluster associations are the paths that have the alternate indexes as path entries. A base cluster could have several associations. The number of associations is limited by the block size of the IAM dataset. For a typical 1/4 track blocked IAM dataset, it can have a maximum of 204 associated datasets.

Alternate Index Associations

For an alternate index, the associations will include the related base cluster, its own self-entry, and any paths that have been defined with this particular alternate index as its path entry.

Path Associations

The path associations will consist of the base cluster, the path, and if the path entry is an alternate index, then the alternate index.

Association Maintenance

In general, dataset management utility software products treat IAM files as non-VSAM datasets. Therefore, these products are not going to know about the associations between these non-VSAM datasets, which will require some additional effort on the part of users. The main areas of concern are renaming one or more of the datasets in an alternate index sphere, or performing a copy or restore to a new name of such a dataset. After changing the name of one or more components of an IAM alternate index sphere, it will be necessary to perform a DEFINE RECATALOG on all of the components affected to insure that the dataset names are all properly reflected in the saved association information. IAM includes special processing on a DEFINE RECATALOG to perform the renames on the internal association data. Procedures for performing this recatalog process are described in Section 10.65 and Section 10.84. There is also an automatic recatalog function that is invoked when an IAM AIX sphere is being opened that will do the recatalog providing that the change in the names of the associations are consistent with the name of the component being opened.

There is support to erase the associations from the base cluster, if for example a base cluster has been copied or restored with a new name. To erase the associations, one can perform a DELETE AIX NOSCRATCH on the base cluster, which will clear out all related alternate index information.
The procedure for defining and using IAM datasets with alternate indexes is identical to the VSAM process. The only difference is to specify on the DEFINE that the base cluster is to be an IAM dataset. This specification is typically done by either adding an OWNER($IAM) parameter to the define parameters, or changing the name to include the $IAM literal. Most applications that are going to be using IAM for their alternate index already have the general procedures established. To convert to IAM, the only change needed is on the define step of the base cluster. An overview of the process for using an IAM alternate index includes the following steps:

- Defining the base cluster in Section 10.20.
- Loading the base cluster with data in Section 10.30.
- Defining the alternate index(es) in Section 10.61.
- Building the alternate index(es), usually with IDCAMS BLDINDEX command in Section 10.62.
- Defining paths to the alternate index(es) and if used, paths to the base cluster in Section 10.63.
- Using Paths to access data in Section 10.64.
- Special Considerations with IAM Alternate Indexes in Section 10.65.

The first two steps of defining and loading the base cluster are described in prior sections the IAM manual. Details and examples of those steps will not be presented here. This section will describe the rest of the procedures for using IAM Alternate Indexes.
10.61 Defining an IAM Alternate Index

Overview

The process of defining an IAM alternate index is virtually identical to defining a VSAM alternate index. IAM will automatically determine if the base cluster is an IAM dataset, so there is no need to specify an OWNER($IAM) on the DEFINE ALTERNATEINDEX. The definition of an alternate index is quite similar to the definition of any other KSDS type of file, with the major difference being the addition of the RELATE(base cluster name) parameter to identify the base cluster. The KEYS parameter will specify the key length of the alternate key, and the position of the alternate key in the base dataset record. The RECORDSIZE parameter specifies the average and maximum size of the records within the alternate index itself. The minimum size for this record will be (5 + alternate key length + base key length). When the base cluster is an ESDS, the base key length will be the RBA length of either 4 or 8.

Two additional alternate index definition parameters are UNIQUEKEY | NONUNIQUEKEY and UPGRADE | NOUPGRADE. The UNIQUEKEY and its opposite NONUNIQUEKEY indicate whether or not there is only one record in the base cluster with any particular alternate key (UNIQUEKEY), or if an alternate key can have multiple records in the base cluster with that key value (NONUNIQUEKEY). The IDCAMS default is NONUNIQUEKEY. The UPGRADE parameter and its opposite, NOUPGRADE, indicate whether IAM will automatically update the alternate index whenever the base cluster is updated in a way that would change the alternate index. The IDCAMS default is UPGRADE.

As with VSAM, an SMS managed alternate index will be given the same STORCLASS and MGMTCLAS as the base cluster. The user can specify a DATACLAS if desired.

The most common method of defining an alternate index is by using the IDCAMS utility program. The name of the base cluster must be included in the alternate index definition, specified by the RELATE parameter. When IAM intercepts an alternate index define, it will look at the base cluster to see if it is an IAM dataset, and if so an IAM alternate index will be defined. Specifying the OWNER($IAM) parameter will force the define process through IAM, which will fail if the base cluster is not an IAM dataset.

The IAM alternate index is an enhanced format IAM KSDS type of dataset, which resides on disk as a non-VSAM file type, with a DSORG of PS set. The data and index component of the alternate index all reside within the single dataset. IAM CREATE overrides can be used when defining an alternate index as described as described in sections 10.20 and 30.02 of this manual.
There is required information that must be provided to define an IAM Alternate Index. Below are the IDCAMS keywords that can be used to provide that information. Please note that you cannot provide an SMS STORCLAS or MGMTCLAS value when defining an Alternate Index. This is a VSAM restriction that the alternate index must be in the same STORCLAS and MGMTCLAS as the base cluster.

<table>
<thead>
<tr>
<th>Essential Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTERNATEINDEX(....)</td>
<td>Identifies that an alternate index dataset is to be created.</td>
</tr>
<tr>
<td>NAME(dsname)</td>
<td>This is a required parameter. The entry name specified for the alternate index will be the name of the IAM dataset. The data and index component names are ignored.</td>
</tr>
<tr>
<td>RELATE(dsname)</td>
<td>Required parameter that specifies the name of the base cluster that this dataset will be an alternate index to. To have an IAM alternate index, the base cluster must be an IAM Enhanced Format KSDS or ESDS type of file.</td>
</tr>
<tr>
<td>KEYS(length offset)</td>
<td>Required parameter that specifies the length of the alternate key, and the relative position of the alternate key within the base cluster. For IAM, the maximum length of the alternate key is 249. Note that each record in the base cluster must be long enough to contain the alternate key.</td>
</tr>
<tr>
<td>RECORDSIZE</td>
<td>Specifies the average and maximum lengths of the records in the alternate index dataset. The minimum record length is 5 plus the alternate key length plus the length of the key for the base cluster. Note for ESDS files the base key length is 4 bytes, or if defined with extended addressability (XESDS), it will be 8 bytes. For an alternate index defined with the UNIQUE attribute, the maximum length will be equal to the minimum length record. For an alternate index defined with NONUNIQUE keys, the maximum length will be based on the maximum number of NONUNIQUE keys there could be for any alternate key. The formula is (5 + alternate key length + (maximum number of base records with same alternate key * primary key length)).</td>
</tr>
</tbody>
</table>

Spanned records will be automatically indicated when the maximum record size exceeds the control interval size. The maximum record size for an IAM file is approximately 8 megabytes. The VSAM architecture allows for up to 32,767 non-unique keys per each alternate key, which IAM abides by to insure full VSAM compatibility.

IDCAMS default values are: average=4086 maximum=32600.
<table>
<thead>
<tr>
<th>Essential Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CYLINDERS(xx yy)</td>
<td>Required information that indicates the amount of DASD space to be allocated for the IAM dataset. The unit of allocation is based on the keyword specified. The first value provided indicates the amount of space to be allocated during the file definition process. For IAM files, the primary quantity MUST be available on the first volume, otherwise the request will fail. The second value, which is optional, indicates the amount of additional DASD space to request in case the primary quantity is insufficient. The secondary quantity will be used to acquire additional extents during the BLDINDEX or reorganizations. The secondary quantity will also be used to acquire additional space as needed to handle updates and inserts after the BLDINDEX. IAM datasets that are not SMS Extended format and are not Large Format are limited to 16 extents per volume. The maximum extent size is 64K-1 tracks (65,535), or 4,369 cylinders per volume, which is also the maximum amount of space that can be used by IAM on any single volume for basic format datasets. To use more than 64k tracks per volume, use either the Large Format datasets, or SMS Extended Format datasets.</td>
</tr>
<tr>
<td>TRACKS(xx yy)</td>
<td></td>
</tr>
<tr>
<td>RECORDS(xx yy)</td>
<td></td>
</tr>
<tr>
<td>MEGABYTES(xx yy)</td>
<td></td>
</tr>
<tr>
<td>KILOBYTES(xx yy)</td>
<td></td>
</tr>
<tr>
<td>VOLUMES(volser....)</td>
<td>Specifies the volume(s) on which IAM is to allocate the dataset. A maximum of 59 volumes can be specified for an IAM dataset, due to z/OS TIOT entry size limitations.</td>
</tr>
</tbody>
</table>
### Optional Parameters

While the following keywords are all optional, in various situations they may be required or quite beneficial. Some of the keywords listed here are not relevant to IAM files, and are presented for that reason. For ease of reference, they are presented in alphabetical sequence.

<table>
<thead>
<tr>
<th>Optional Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUFFERSPACE(bytes)</td>
<td>Specifies the maximum amount of virtual storage to be used for buffers for this dataset. IAM will use this value to calculate the effective MAXBUFNO for accessing the dataset, providing it does not go below the default value from the IAM Global Options Table.</td>
</tr>
<tr>
<td>CONTROLINTERVALSIZE (size)</td>
<td>For VSAM, this controls the logical and physical block size on DASD for the file. IAM will refer to this value when calculating the block size it is going to use for the dataset, generally increasing the value to one that will maximize utilization on the device type to which the dataset is allocated. IAM does store the specified CI SIZE, so that it is available information if the dataset is converted back to VSAM.</td>
</tr>
<tr>
<td>DATACLAS(dataclass)</td>
<td>For SMS installations, this parameter specifies the name of the SMS DATA CLASS construct, which provides the allocation attributes for the new dataset. Attributes from the DATACLAS will be used, unless otherwise explicitly specified on the DEFINE statement. Values provided by a Data Class include maximum record size, key length and key offset, space allocation values, free space, share options, CI size, and volume count.</td>
</tr>
<tr>
<td>FILE(ddname)</td>
<td>Optional keyword that specifies a DDNAME that allocates the volume(s) on which the IAM file is to be allocated. This keyword can also be used to relate a particular IAM CREATE Override statement with a matching DDN= specification.</td>
</tr>
<tr>
<td>FREESPACE( CI%, CA%)</td>
<td>Specifies the amount of space to be reserved for future inserts or updates when the file is being loaded. <strong>CI%</strong>: Specifies the amount, as a percentage, of space to be left available in each prime block of the IAM file. Unless a file is never updated and never has records added to it, some CI free space should be specified. This is of particular importance to data compressed files that are updated, because even if the application does not change the length, the stored record may end up being longer after compression. For most datasets, CI free space should be kept very small, with 0 being generally recommended, otherwise a lot of dasd space can be wasted with capacity that is not going to be used. <strong>CA%</strong>: Controls how much DASD space is released at the end of a file load. Using ½ of the specified percentage, a target amount of DASD space to be reserved for future expansion is computed. If the amount of available DASD space within the file extent(s) is equal to or less than the amount to be reserved, then no space is released. IAM will not acquire additional extents to meet the space reservation. If the amount of DASD space exceeds the reserved value, then the excess will be released.</td>
</tr>
</tbody>
</table>

If your installation has the DSORG=DA set, then be aware that z/OS will not allow any space to be released.
Defining an IAM Alternate Index

**Optional Keyword** | **Description**
--- | ---
MODEL(dataset name) | Specifies that the attributes of the dataset being defined are to be copied from an existing VSAM or IAM alternate index. This capability is only relevant for basic file characteristics, such as record size, key length and offset, space allocation values, volumes, and free space values. Any IAM Overrides are NOT picked up by the MODEL parameter. Likewise, any VSAM file attributes that are ignored by IAM are not available either, such as IMBED, SPEED, REPLICATE, KEYRANGES, etc.

If you are trying to define a VSAM AIX using an IAM AIX as a model, you MUST provide an OWNER parameter, with a value that does not contain $IAM.

NOTE: When using MODEL with ANYVOL, the SUBALLOCATION parameter must also be specified. Although it is ignored by IAM, it prevents IDCAMS from trying to allocate ANYVOL.

OWNER($IAM) | Optional keyword that indicates that the alternate index is to be an IAM dataset. IAM will automatically assume that an alternate index is to be an IAM dataset when the base cluster is also an IAM dataset. If no value has been specified for OWNER, IAM will set it to $IAM in the catalog entry.

RECATALOG | Is an optional keyword for existing IAM datasets to reestablish the catalog entry, and update the internal alternate index relationship information. Requires the user to specify the dataset name, the volume(s) on which the dataset resides or ANYVOL, and the name of the related base cluster. Recatalog is also used after renaming one or more components of an alternate index sphere, to reset the associated dataset names to the new names.

REUSE | Specifies whether the file being defined can be reloaded (or reorganized) without being redefined. IAM defaults to REUSE, which is that any IAM file can be reloaded without having to be deleted and redefined. To use this feature with IDCAMS REPRO, specify the REUSE keyword.

NOREUSE | IAM does provide a Global Option, ENABLE=NOREUSE, which if set will cause IAM to honor the specification of REUSE or NOREUSE. If that Global Option has been set, then IAM will honor the NOREUSE setting just like VSAM. While quite rare, there are a few application programs that rely on the NOREUSE setting.

NOREUSE will not allow a file to be reloaded without being deleted and redefined. An exception to this is made if the program issuing the OPEN is FDRREORG, in which case it will be allowed. If any other attempt is made to do so, the OPEN will fail with a return code of 8, and the ACB error flag set to 232, or x'E8'.
### Optional Keyword

**SHAREOPTIONS**

<table>
<thead>
<tr>
<th>cross-region</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(cross-system)</td>
<td>Specifies the level of protection provided by the access method to prevent or allow sharing of data within the file. The protection mechanisms include the z/OS ENQ service, and the internal IAM buffering techniques.</td>
</tr>
</tbody>
</table>

The first parameter specifies how a file can be shared in the same system (CPU). The second parameter specifies how a file is shared between systems.

**NOTE**: With IAM/RLS or IAM/PLEX, IAM supports automatic record level sharing for concurrent users of the IAM dataset. When IAM/RLS and IAM/PLEX are not active, IAM supports the cross-region share options with the z/OS ENQ service, the same as VSAM. IAM does not support the cross-system share options. IAM issues an ENQ with a major name (QNAME) of IAMENQ and the dataset name plus first volume serial as the minor name (RNAME). If you need to enforce ENQ protection cross-system then you must add the major name of IAMENQ to your CA-MIM or GRS control files or whatever ENQ control product you use.

**Cross Region Share Option Values:**

- **1** Any number of users for read OR one user for update. The file’s structure, data integrity, and read integrity are fully preserved.

- **2** Any number of users for read AND one user for update. The file’s structure and data integrity are fully preserved. *If the file is currently opened for update, other users reading the file do not have read integrity unless files are processed with IAM/RLS or IAM/PLEX*. They may not be able to access inserted or updated records, if such records were added to the overflow areas of the file, without closing and reopening the dataset.

- **3** Any number of users for read or update and users are responsible for integrity. Updated blocks are immediately written back out to DASD. *Use of this share option for IAM files is strongly discouraged unless the file is shared using IAM/RLS, IAM/PLEX, or some other VSAM sharing software. Due to the nature and structure of the index to the IAM overflow area, the data integrity of IAM files is compromised by use of this share option value.*

- **4** Any number of users for read or update, and users are responsible for integrity. IAM will use only a single buffer, and each logical I/O request will cause the buffer to be refreshed, and subsequently rewritten if the record is updated. *Use of this share option for IAM files is strongly discouraged unless the file is shared using IAM/RLS, IAM/PLEX, or some other VSAM sharing software. Due to the nature and structure of the index to the IAM overflow area, the data integrity of IAM files is compromised by use of this share option value.*

**SUBALLOCATION**

IAM files are always allocated as if they were unique clusters. However it may be necessary to specify this parameter when using the IAM non-specific allocation (ANYVOL) and the MODEL parameter. SUBALLOCATION will prevent IDCAMS from allocating the volumes indicated in the VOLUME parameter when the MODEL parameter is specified.
Defining an IAM Alternate Index

<table>
<thead>
<tr>
<th>Optional Keyword</th>
<th>Description</th>
</tr>
</thead>
</table>
| TO(date) FOR(days) | Specifies the retention period for the file being defined. This parameter has the same meaning for an IAM file as a VSAM file. The expiration date is placed in the VTOC for the dataset, and in the catalog entry. The keyword PURGE must be specified on the DELETE to cause the file to be scratched.  
TO(date) - gives the date in the form YYYYDDD (four or two digit year and three digit Julian date), through which the IAM file defined is to be date protected.  
FOR(days) - gives the number of days up to 9998, through which the IAM file being defined is to be date protected. A value of 9999 results in permanent retention.  
Default: Dataset is not date protected. |
| UNIQUE | This parameter has no relevance for IAM files, as IAM files are always unique.  
However, the user should be aware that specification of this keyword results in IDCAMS allocating the specified volumes prior to issuing the actual define request. For this reason, it is recommended that this parameter not be specified for IAM files.  
If the customer is using the IAM non-specific device allocation, i.e. ANYVOL, then this parameter must not be specified. |
| UNIQUEKEY | Specifies whether more than one record can have the same alternate key.  
When UNIQUEKEY is specified, each alternate key can only index one data record. With NONUNIQUEKEY, each alternate key can index one or more data records. The number of non-unique keys that can be contained in a single record is limited by VSAM architecture to 32,767.  
Default is NONUNIQUEKEY. |
| NONUNIQUEKEY |  |
| UPGRADE | Specifies whether IAM will upgrade (update) the alternate index as the base cluster is modified.  
UPGRADE indicates that IAM will automatically update the alternate index whenever records are updated, inserted, or deleted from the base cluster.  
NOUPGRADE indicates that IAM will not automatically update the alternate index.  
Default is UPGRADE. |
Example A: Basic Alternate Index Define

Shown below is an example of using IDCAMS to define an alternate index for an IAM base cluster. For ease of conversion from VSAM to IAM, the need to supply an OWNER($IAM) on the define request has been eliminated for alternate indexes and paths. When defining the alternate index, if the base cluster is a VSAM dataset, then the define request is passed to VSAM, as long as OWNER($IAM) is not specified. If the base cluster is an IAM dataset, then IAM will automatically define the alternate index as an IAM dataset.

```
//AIXDEFIN EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//SYSIN DD * 
DEFINE ALTERNATEINDEX -
  (NAME(example.iam.aix) -
   RELATE(example.iam.cluster) -
   KEYS( len offset) -
   RECORDSIZE(avg   max) -
   SHAREOPTIONS(2 3) -
   FREESPACE( ci% ca%) -
   VOLUMES(myvol1) -
   CYLINDERS( primary secondary) -
   NONUNIQUEKEY -
   UPGRADE)
   LISTCAT ENT(example.iam.aix) ALL 
/
```

Figure 10-38: Example of defining an IAM Alternate Index (EX1061A)
Defining an IAM Alternate Index with IAM ISPF Panels

Customers that have installed the IAM ISPF panels are able to define the IAM Alternate Index datasets using the IAM ISPF panels. The panels feature a fill in the blanks mechanism, along with being able to provide various relevant IAM overrides. After displaying the IAM Primary Option Menu, select option I for defining an IAM dataset. Be sure to fill in the dataset name of your alternate index using standard TSO naming conventions. Indicate that you are defining an alternate index by placing an X in the dataset type field. Optionally, you can specify an existing IAM or VSAM alternate index as a model for the dataset attributes. Below is an example of the IAM Primary Option Menu with parameters set for defining an IAM alternate index.

![IAM Primary Options Panel set for AIX Definition]

Figure 10-39: IAM Primary Options Panel set for AIX Definition
When you hit enter, the IAM Alternate Index Definition panel is displayed. If you had specified a MODEL dataset, the information from the model would be filled in on the definition panel. You can set or change any of the attributes by filling in the blanks. Note that the fully qualified dataset name of the alternate index is displayed at the top left corner. Just below the alternate index name is a place for you to fill in the fully qualified name of the base cluster for this alternate index.

![Figure 10-40: Define an IAM Alternate Index ISPF Panel](image)

In the above example, the information displayed in red represents what was provided by the user on this panel. Once all of the information is complete, you can hit enter, and IAM will allocate the alternate index. When allocation is completed, IAM will automatically return to the IAM Primary Option Panel, with the message DATASET ALLOCATED in the upper right hand corner. By hitting enter again you can display the newly defined alternate index’s attributes.
IAM responds with the IAM Alternate Index File Characteristics display, as shown below. The various attributes, where applicable, are filled in. At the bottom left is a section labeled Associations. This will contain related base cluster and the alternate index dataset itself. Once one or more paths have been defined for this alternate index, they will also be included under the associations.

Figure 10-41: Figure 38: IAM Alternate Index File Characteristics Panels
10.62 Building an IAM Alternate Index

The process of building an IAM alternate index is identical to the VSAM process. The alternate index is usually built by use of the IDCAMS BLDINDEX command. However, other programs that have been written to build alternate indexes for VSAM can be used to build them for IAM. The internal record format of the IAM alternate index record is identical to the VSAM format.

Using IDCAMS BLDINDEX

The BLDINDEX command for IDCAMS has two main operands, specifying the input file, or dataset, which is the base cluster, and specifying the output file or dataset, which is the alternate index. As long as there is sufficient virtual storage to sort the alternate key / prime key pairs, or if you have DFSORT, the example shown below of using IDCAMS BLDINDEX can be used. For complete information on using the IDCAMS BLDINDEX please refer to the IBM manual "DFSMS/MVS ACCESS METHOD SERVICES for the Integrated Catalog Facility", number SC26-4906.

```bash
//BLDINDEX EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//IAMINFO  DD   SYSOUT=*  
//BASE     DD   DISP=OLD,DSN=example.iam.cluster
//AIX      DD   DISP=OLD,DSN=example.iam.aix
//SYSIN    DD   *

BLDINDEX INFILE(BASE)OUTFILE(AIX)
/*
```

Figure 10-42: Example of IDCAMS BLDINDEX (EX1062A)

There are a few things to notice on the above example. First, the base cluster is allocated with a disposition of OLD. This is done to prevent updates from occurring while the alternate index is being built that could cause an out of sync condition with the alternate index. Second, there is an optional IAMINFO DD card in the JCL. This will cause IAM to produce an activity report for each IAM dataset that has been used.
If you do not have DFSORT and do not have enough virtual storage available for an internal sort, then you will need to add two work files for the BLDINDEX to work. The space requirements will depend on the size of your files, and up to five volumes can be specified for each work file. IDCAMS will define two ESDS type of files to these DD statements to use for sorting the data. The default DD names used by IDCAMS are IDCUT1 and IDCUT2. An example is shown below of using the work files.

```plaintext
//BLDINDEX EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=* 
//IAMINFO DD SYSOUT=* 
//IDCUT1 DD DSN=workfile1,VOL=SER=volser, UNIT=unitname,DISP=OLD,AMP='AMORG'
//IDCUT2 DD DSN=workfile2,VOL=SER=volser, UNIT=unitname,DISP=OLD,AMP='AMORG'
//BASE DD DISP=OLD,DSN=example.iam.cluster 
//AIX DD DISP=OLD,DSN=example.iam.aix 
//SYSIN DD * 
   BLDINDEX INFILE(BASE)OUTFILE(AIX) /*
```

Figure 40: Example of IDCAMS BLDINDEX with work files (EX1062B)
The IAM ISPF panels offer a build index function as well, that will invoke the IDCAMS BLDINDEX function. To go to the build index function from the IAM Primary Options panel select option U for the utilities panel. From the IAM Utility Program Menu, as shown below, select option I to build the alternate index.

Figure 10-43: IAM ISPF Utility Selection Menu

After hitting enter IAM will display the Build Index panel. On this panel, enter the name of the base cluster and the name of the alternate index you want to build. After entering that information, press the enter key.

Figure 10-44: Example of IAM ISPF Build Index Panel

When the Build Index function is complete, IAM ISPF will open a browse panel of the output from the IDCAMS BLDINDEX function. An example is shown below:
Figure 10-45: Sample Output from BLDINDEX Function
Defining an IAM Path

Defining a Path

The process of defining a PATH to an IAM alternate index or base cluster is essentially the same as with VSAM. IAM will automatically determine if the PATHENTRY is an IAM dataset, so use of the OWNER($IAM) parameter is optional. An IAM path is a one-track dataset. The allocation request will be made for the same volume on which the path entry dataset resides. If the path entry dataset is DFSMS managed, the path dataset will be allocated with the same SMS classes as the path entry dataset. IAM uses the one track dataset to store information about the relationship of the path to its alternate index or base cluster, rather than using a catalog entry. The IAM path dataset does not have to be on the same volume with the related dataset, although IAM attempts to place it there.

IDCAMS Define Path

IDCAMS can be used to define a path. On the DEFINE PATH command, the PATHENTRY keyword indicates the related dataset. To access a base cluster through an alternate index, the PATHENTRY will be the name of the alternate index. The attribute UPDATE (or NOUPDATE) indicates whether the upgrade set of the alternate indexes will also be opened. The upgrade set includes all alternate indexes defined with the UPGRADE attribute.

If only the base cluster is being opened for update, any upgradeable alternate index datasets will also be opened. To prevent the alternate index upgrades from being done, define a PATH for the base cluster with the NOUPDATE attribute.

An example of defining an IAM PATH with IDCAMS is shown below.

```
//DEFPATH EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*                  
//SYSPRINT DD SYSOUT=*                  
//SYSSIN DD *                           
DEFINE PATH  
   (NAME(example.iam.path) - 
   PATHENTRY(example.iam.aix) - 
   UPDATE )                        
/*
```

Figure 10-46: Example of Using IDCAMS to Define a Path (EX1063A)
If you have the IAM ISPF panels installed, they also provide a Define Path function. Starting at the IAM Primary Options menu, as shown below, select option I to define an IAM dataset then enter the name of the path that you want to define, and indicate a P for dataset type, indicating a path. An example of the completed panel is shown below.

![IAM ISPF Define Path Panel](image)

Figure 10-47: IAM ISPF Define of a Path

After hitting enter, IAM will display the IAM Define Path panel as shown below. On this panel, enter the fully qualified name of the path entry dataset, which is either an alternate index or a base cluster. Also enter whether or not the upgrade set is to be opened if the path is opened for update. While a specification of NO will eliminate the overhead of updating all of the alternate indexes in the upgrade set, the alternate indexes will not properly reflect the contents of the base cluster.

![IAM ISPF Define an IAM Path Panel](image)

Figure 10-48: IAM ISPF Define an IAM Path Panel

After hitting enter IAM will return to the IAM Primary Options panel, with a message in the upper right hand corner indicating that the path was successfully allocated. To display the attributes of the Path, hit enter again, with the Path name in the dataset name field.
IAM Path Display

The IAM Path Characteristics panel displays basic information about the actual data record that would be retrieved from the base cluster through this path. Because the IAM Path dataset is an actual one track dataset, volser and device type information is displayed. At the bottom are the associations indicating the datasets related to this path.

---

**Figure 10-49: IAM Path Characteristics Panel**
10.64 Accessing Data Through an IAM Alternate Index

Now that you have a base cluster defined and loaded, an alternate index defined and built, and a path defined, you are ready to start accessing your data via the path. Just as with VSAM, all you need to do is to allocate (dynamically or with a DD card) the path to the alternate index, open it and you are ready to go. When your program opens the path, IAM will dynamically allocate and open all of the necessary components of the alternate index sphere. For example, if you were to just be opening the path for read only access, IAM will dynamically allocate the alternate index and the base cluster datasets, open them, and set up the control blocks and routines necessary for your program to access the data. If the program opens up the path for update, then any associated upgradeable alternate indexes will also be dynamically allocated and opened. IAM will use the system generated DD names for the files that are dynamically allocated, which have the format SYSnnnnn. When you close the path, IAM will close and deallocate all of the component datasets that had been opened.

COBOL programmers, and users of programs written in the COBOL language are encouraged to review the COBOL Programming Guide for information on using alternate indexes and paths with COBOL programs. COBOL requires certain conventions for the DD names used in these circumstances that must be followed for proper functionality. The requirements that COBOL has for VSAM datasets must be followed with IAM datasets as well.

Because the DD names used to access the component IAM datasets in the alternate index sphere cannot be known in advance, the IAM override processor has been enhanced to handle specification of overrides by dataset name (DSN=) in addition to the DD name. Using the DSN= parameter on an IAM ACCESS override is valid for any enhanced format IAM file. The DD= override can still be specified on other override cards, which can be intermixed with the new DSN= parameter. Information on the IAM ACCESS overrides is in Section 10.40 or Section 30.03. An example of using a path, along with IAM ACCESS overrides by DSN= is shown below:

```
//UPDATAIX EXEC PGM=mypgm
//PATH       DD DISP=SHR,DSN=example.iam.path
//IAMINFO    DD SYSOUT=*  
//IAMOVRID   DD *
ACCESS       DSN=example.iam.base,MAXBUFNO=32  
ACCESS       DSN=example.iam.aix,DYNDS=32
/*
```

Figure 10-50: Example of JCL to use a Path with IAM Override
10.65 Special Considerations with IAM Alternate Indexes

Renaming and Copying

There are special considerations when any associated dataset or all of the associated datasets in an IAM alternate index sphere are renamed. These same considerations exist if the datasets are copied or restored with a new name. Because IAM datasets appear to the utilities performing such functions as non-VSAM datasets, the association information is not automatically updated when a dataset name is changed. To update the association information, a DEFINE RECATALOG must be done on all of the affected alternate index and path datasets.

IAM has a function that will automatically perform this recatalog process when the renamed dataset is opened, providing that the change in the name is identical in each of the cluster, path, and alternate index components. IAM will determine the new name of each of the components and first verify that all of the components had identical name changes, then if so it will internally perform the recatalog. If it fails, then a manual recatalog will need to be performed.

To do the manual recatalog, the base cluster should be recataloged first, followed by the alternate index dataset(s), and then the path datasets. The information that must be provided includes the new dataset name, the new related or path entry dataset name, and for base clusters and alternate index datasets, the volume(s). If the volume(s) are unknown or include DFSMS candidate volumes, then specify a volume of ANYVOL. The ANYVOL specification will indicate that IAM will use the volume(s) that are currently in the catalog. The base cluster RECATALOG must also include the OWNER($IAM) parameter, which is optional for the Alternate Index and Path recatalog. The IAM alternate index support intercept for the DEFINE RECATALOG function has processing to reset the association names. An example of a DEFINE RECATALOG is shown below.

```
//RECAT    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*  
//SYSIN    DD   *
DEFINE CLUSTER- 
    (NAME(newname.iam.cluster)- 
     OWNER($IAM)- 
     VOLUME(ANYVOL)- 
     RECATALOG )
DEFINE ALTERNATEINDEX- 
    (NAME(newname.iam.aix)- 
     RELATE(newname.iam.cluster)- 
     VOLUMES(ANYVOL)- 
     RECATALOG )
DEFINE PATH- 
    (NAME(newname.iam.path)- 
     PATHENTRY(newname.iam.aix)- 
     RECATALOG)
LISTCAT ENT(newname.iam.cluster) ALL  
LISTCAT ENT(newname.iam.aix) ALL  
LISTCAT ENT(newname.iam.path)ALL
/*
```

Figure 10-51: Example of Recatalog after a Rename (EX1065A)
Removing Association Information

If you have only made a copy of the base cluster and renamed it, you can eliminate the association information from the base cluster by issuing a DELETE cluster name AIX NOSCRATCH. IAM will recognize this special delete and remove all of the associations from its definition. This delete must be done prior to performing a recatalog as described above, because IAM checks before erasing the association information that the current base cluster name does not match the base cluster name in the associations.

Delete Support

IAM alternate index support includes special processing for dataset deletion. As with VSAM, when the base cluster is deleted, all related alternate index and path datasets will also be deleted, providing that the SVC 26 DELETE interface is used. This interface also requires that the related datasets be cataloged. If a related dataset is not cataloged, it will be skipped over by the delete process. This is the interface that is used by IDCAMS, the TSO DELETE command, and the IAM ISPF panels. If the SCRATCH SVC interface is used for non-SMS managed volumes, then IAM will not intercept the request, and only the specified dataset will be deleted.

If a delete is issued for an alternate index, then the alternate index and any related paths will be deleted. If the delete is issued for a path, only the path will be deleted. These actions are consistent with VSAM delete processing.
IAM Reports

IAM offers a variety of reports to assist in the management of IAM datasets. The two primary reports are the IAMPRINT report and the IAMINFO report. The IAMPRINT report is automatically produced when a LISTCAT ALL is done on an IAM dataset. The IAMINFO report is a run time report that is produced whenever an IAM dataset is closed, providing that there is an IAMINFO DD card allocated to the job. The IAMINFO reports can also be generated from the IAM SMF records, if they are collected, using the IAMSMF program. The IAMSMFVS program produces a more concise set of reports, based on an accumulation of the IAM SMF records for each dataset. There are three different reports produced by IAMSMFVS, all of which have one line per dataset.
10.71 IAMPRINT Report

LISTCAT Report
The IAMPRINT report is produced by an IDCAMS LISTCAT ALL command. This report contains information about the dataset attributes, the DASD volumes being used, along with the amount of space on each volume, plus various statistics about the dataset. The statistics that are presented in the IAMPRINT report are generally only updated when the dataset is closed. If the dataset is currently open for update the statistics may not accurately reflect the current status of the dataset. A message will appear at the bottom of the report if this is determined to be true by the IAM LISTCAT processor. The statistics can also be invalid due to a system failure that prevented the dataset from being properly closed, or if the dataset was being concurrently shared by multiple jobs for update processing, without going through IAM RLS. IAM does not support or provide for data integrity when a dataset is shared for concurrent update purposes unless IAM/PLEX, IAM/RLS or other software that enables sharing is used. Doing so may result in the loss of data.

LISTCAT Example
Specification of the IAMPRINT DD is optional. If the IAMPRINT DD is not specified, then IAM will allocate one to SYSOUT=* for batch jobs, providing that the SYSPRINT DD is also allocated to a SYSOUT class. For TSO users, the IAM output will be displayed directly on the screen. In both cases, the report will be produced on the IAMPRINT DD dataset, if the file was explicitly specified by the user. The IAM LISTCAT processing can be disabled by placing a //IAMNOLIC DD DUMMY card in the job step performing the LISTCAT(s). Using the IAMNOLIC DD card is intended for jobs that are performing generic LISTCAT operations, and the IAM information is not necessary.

```
//LISTCAT EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*  
//SYSIN    DD   *  
LISTCAT ENT(IAMV.VAR100.CLUSTER) ALL
/*
```

Figure 10-52: JCL to Obtain IAM Listcat (IAMPRINT) Output (EX1071A)
LISTCAT Output  The results of running the above job on an IAM dataset are two printed reports. The first report is produced by IDCAMS on SYSPRINT that displays the information on the actual non-VSAM catalog entry for the IAM dataset. The second report is produced by IAM on IAMPRINT that displays the attributes and other information about the dataset itself. The two reports are shown below.

Figure 10-53: IDCAMS LISTCAT Output from SYSPRINT

As you can see, the IDCAMS output indicates that the dataset type is non-VSAM. While the OWNER field is set to '$IAM', it is not a reliable indicator that the dataset is in fact an IAM dataset.
Below is an example of the IAMPRINT Report for an Enhanced Format IAM dataset. The fields displayed on the report will vary, depending on the type and status of the IAM file.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE FORMAT</td>
<td>ENHANCED</td>
</tr>
<tr>
<td>RECORD SIZE</td>
<td>2040</td>
</tr>
<tr>
<td>CI SIZE</td>
<td>2048</td>
</tr>
<tr>
<td>BLOCK SIZE</td>
<td>13682</td>
</tr>
<tr>
<td>BLOCK FACTOR</td>
<td>4</td>
</tr>
<tr>
<td>VAR OVERFLOW</td>
<td>YES</td>
</tr>
<tr>
<td>KEY SIZE</td>
<td>58</td>
</tr>
<tr>
<td>FILE TYPE</td>
<td>KSDS</td>
</tr>
<tr>
<td>DEVICE TYPE</td>
<td>3390</td>
</tr>
<tr>
<td>VOLSER</td>
<td>SCR092</td>
</tr>
<tr>
<td>TOTAL EXTENTS</td>
<td>2</td>
</tr>
<tr>
<td>PRIMARY SPACE</td>
<td>500</td>
</tr>
<tr>
<td>MULTIVOLUME</td>
<td>PRIMARY</td>
</tr>
<tr>
<td>RELEASE</td>
<td>NO</td>
</tr>
<tr>
<td>DATA COMPRESS</td>
<td>HARDWARE</td>
</tr>
<tr>
<td>DICTIONARY</td>
<td>*DYN</td>
</tr>
<tr>
<td>TOTAL RECORDS</td>
<td>391000</td>
</tr>
<tr>
<td>UPDATE</td>
<td>0</td>
</tr>
<tr>
<td>HIGH USED RBA</td>
<td>322895200</td>
</tr>
<tr>
<td>FILE DEFINED</td>
<td>2014.294</td>
</tr>
<tr>
<td>FILE LOADED</td>
<td>2014.294</td>
</tr>
<tr>
<td>LAST UPDATED</td>
<td>2014.294</td>
</tr>
<tr>
<td>STORAGE REQUIRED</td>
<td>1113392</td>
</tr>
<tr>
<td>NUMBER OF IAM DATA BLOCKS</td>
<td>21899</td>
</tr>
<tr>
<td>EXTENDED HIGH ALLOCATED RBN</td>
<td>29219</td>
</tr>
</tbody>
</table>

Figure 10-54: Example of IAMPRINT Output of an Enhanced Format Dataset
The IAMPRINT output for an alternate index is almost identical to the report for a base cluster. An IAM Alternate Index is essentially an IAM KSDS type of file. The file type indicates that this is an AIX (Alternate Index) dataset. Just below the DATA COMPRESS line is an additional line that includes the alternate key offset in the base record, and to the right of that is the indication of whether or not the alternate index keys are unique. At the bottom of the report are the associations. The associations will also appear on a base cluster once there are defined alternate indexes and paths to that base cluster.

**Figure 10-55: Example of IAMPRINT for an IAM Alternate Index**
The report for an IAM Path dataset is similar in format to the above report, only much shorter. An IAM Path is a 1-track dataset, which is used to provide access to a base cluster through an alternate index, or to the base cluster directly. The information of most interest and concern in this report is the associations. The Path serves as a connector that is used by OPEN processing to determine what datasets need to be allocated and opened.

The file attributes displayed, including record size, alternate key size, and alternate key offset will all reflect the data record as retrieved through the path. The record size will always be the base cluster maximum record size. If the Path is for an alternate index, then the alternate key size and alternate key offset will be as defined for the alternate index. If the Path is for a base cluster, then the alternate key size and alternate key offset will be the base cluster’s key size and offset.

An example of an IAMPRINT report on a PATH is shown below.

![IAMPRINT report example](image)

*Figure 10-56: Example of an LISTCAT Output for an IAM Path*
The fields on the IAMPRINT report for an Enhanced format file are described below. The table starts with the fields on the left side of the report, and is followed by all of the fields on the right side of the report. The last portion describes the fields that appear as a single column at the bottom of the report. Note that not all fields are present in all variations of the report.

The header of the report has an IAM400 message followed by the IAM100 message. The message numbers are not displayed when the output is directed at a TSO terminal. The IAM400 message indicates that the report is from a catalog information request (i.e., LISTCAT) along with the version and level of IAM producing the report. The date and time of the report is included in that message. The dataset name is indicated on the IAM100 message.

<table>
<thead>
<tr>
<th>IAMPRINT Field Descriptions</th>
<th>Description of Field Contents:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Left Column:</strong> File Format</td>
<td>Indicates the file format use to store the data. Possible values are:</td>
</tr>
<tr>
<td><strong>File Format</strong></td>
<td><strong>ENHANCED</strong> – Indicates that the file is an enhanced format IAM file.</td>
</tr>
<tr>
<td></td>
<td><strong>LARGE-FRMT</strong> – Indicates an Enhanced format IAM file that is stored in a Large Format sequential dataset meaning that it can exceed 64K tracks per volume.</td>
</tr>
<tr>
<td></td>
<td><strong>SMS-EXT</strong> – Indicate that the file is a SMS Extended format and an IAM enhanced format file.</td>
</tr>
<tr>
<td></td>
<td><strong>COMPATIBLE</strong> – Indicates that the file is a compatible format file, which can be used with the older non-VSAM interfaces to IAM.</td>
</tr>
<tr>
<td><strong>Record Size</strong></td>
<td>The maximum amount of data that can be contained in a single record. Value is from the RECORDSIZE parameter on the IDCAMS DEFINE statement.</td>
</tr>
<tr>
<td><strong>CI Size</strong></td>
<td>The specified Control Interval (CI) size from the DEFINE. If CI size was not specified on the Define, IAM calculates a valid VSAM CI size based on the maximum record size.</td>
</tr>
<tr>
<td><strong>Block Size</strong></td>
<td>The physical block size IAM is using for the file. A block is the amount of data transferred in an I/O operation, and that is stored in a single contiguous stream of data on the actual device.</td>
</tr>
<tr>
<td><strong>Block Factor</strong></td>
<td>Indicates the number of blocks per track, or the user specified block size from an IAM CREATE Override.</td>
</tr>
<tr>
<td><strong>Var Overflow</strong></td>
<td>Indicates if IAM is using variable length overflow for this file. If the value is <strong>YES</strong>, IAM will put as many records as can fit within each overflow block. If the value is <strong>PRO</strong>, then IAM is using the Prime Related Overflow format, which is a variable length overflow. If the value is <strong>NO</strong>, IAM will only place in each overflow block the number of maximum length records that will fit in an overflow block.</td>
</tr>
<tr>
<td><strong>Key Size</strong></td>
<td>For KSDS type of files, indicates the defined length for the key of each data record.</td>
</tr>
<tr>
<td><strong>Key Offset</strong></td>
<td>Specifies the relative position of the key, as an offset from the beginning of the record, where a value of 0 indicates the first byte. For an AIX type of dataset, this is the key offset within the alternate index dataset itself, and will always be 5.</td>
</tr>
<tr>
<td><strong>Left Column:</strong></td>
<td><strong>Description of Field Contents:</strong></td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>File Type</td>
<td>Indicates the file type. Possible values are:</td>
</tr>
<tr>
<td></td>
<td><strong>KSDS</strong> – Keyed sequence dataset</td>
</tr>
<tr>
<td></td>
<td><strong>ESDS</strong> – Entry sequence dataset.</td>
</tr>
<tr>
<td></td>
<td><strong>ESDS/P</strong> – Entry sequence dataset with PSEUDORBA.</td>
</tr>
<tr>
<td></td>
<td><strong>ESDS/EA</strong> – Entry sequence dataset with 8-byte RBA values.</td>
</tr>
<tr>
<td></td>
<td><strong>AIX</strong> – An IAM Alternate Index dataset</td>
</tr>
<tr>
<td></td>
<td><strong>PATH</strong> – An IAM Path dataset</td>
</tr>
<tr>
<td></td>
<td><strong>RRDS</strong> – Fixed length record RRDS</td>
</tr>
<tr>
<td></td>
<td><strong>VRRDS</strong> – Variable length record RRDS</td>
</tr>
<tr>
<td>Device Type</td>
<td>Indicates the type of DASD device architecture on which the dataset currently resides, for example a 3390.</td>
</tr>
<tr>
<td>EATTR</td>
<td>If the file was defined with an EATTR attribute this line is printed and the value appears here. Possible values are <strong>OPT</strong> or <strong>NO</strong>.</td>
</tr>
<tr>
<td>Volume Count</td>
<td>The number of volumes contained in the catalog entry for this file.</td>
</tr>
<tr>
<td>Volser</td>
<td>This indicates the volume(s) to which the dataset is cataloged. This line is repeated for each volume that is in the catalog entry for the file, except for SMS candidate volumes.</td>
</tr>
<tr>
<td>Total Extents</td>
<td>Indicates the total DASD extents allocated to the file. An extent is a contiguous area of space on the device being used by the dataset.</td>
</tr>
<tr>
<td>Primary Space</td>
<td>Indicates the requested primary space quantity that was specified when the file was originally defined.</td>
</tr>
<tr>
<td>Multivolume</td>
<td>Has a value of <strong>PRIMARY</strong> or <strong>SECONDARY</strong>, which indicates from which space parameter the size of the first extent on each DASD volume is allocated.</td>
</tr>
<tr>
<td>Release</td>
<td>Indicates if DASD space will be released when the file is loaded again. Generally this is <strong>YES</strong> before a file has been loaded, and is set to <strong>NO</strong> after the first load.</td>
</tr>
<tr>
<td>Data Compress</td>
<td>Indicates whether this file can contain IAM data compressed records. Possible values are:</td>
</tr>
<tr>
<td></td>
<td><strong>YES</strong> – Indicates file is eligible for IAM software compression.</td>
</tr>
<tr>
<td></td>
<td><strong>HARDWARE</strong> – Indicates file is eligible for IAM use of the IBM hardware compression function.</td>
</tr>
<tr>
<td></td>
<td><strong>NO</strong> – No data compression will be performed for this file.</td>
</tr>
<tr>
<td>Dictionary</td>
<td>Indicates the name of the hardware compression dictionary that the user has specified for this file. This is only present when the file is enabled for hardware compression. A value of <strong>DYN</strong> indicates that the dictionary was dynamically built by IAM during the load of the file. Any other value indicates the name of customized hardware compression dictionary.</td>
</tr>
</tbody>
</table>
### Description of Field Contents:

**RLSID**
Displays the four character name of the preferred RLSID of an IAM RLS address space for this dataset. This field is only displayed if the dataset was DEFINED or LOADED with the IAM CREATE Override of RLSID. If the displayed RLSID is prefixed with an "*", then it is displaying the value of the RLSID from the current Global Options Table, and indicates that the RLSID keyword was specified on the CREATE override with no value.

**AIX Key Offset**
For an Alternate Index type of dataset, indicates the offset of the alternate key within the base cluster.

**Total Records**
The number of user data records in the file, as of the last close.

**Updates**
The number of user data records that have been updated since the file was last loaded.

**High Used RBA**
For IAM KSDS files, indicates the amount of space used for the file, in bytes. For IAM ESDS files, indicates the amount of actual user data contained within the file.

**IAM Journal**
Indicates that IAM journaling has been specified, either through the JRNAD=IAM override, or through the LOG() IDCAMS define keyword. Possible values are ENABLED indicating that IAM journaling is active, or NONE indicating journaling has been explicitly disabled for this file.

**File Status**
Indicates if the file is LOADED or UNLOADED. A file will be in the UNLOADED state if it has been defined but has not had a successful load, or if a file load or reorganization has failed or is in progress.

**Avg Record Size**
Value is the average record size specified when the file was defined, which may not be equal to the actual stored average record size.

**Freespace - CI% - CA%**
The amount of CI% and CA% free space specified on the file definition.

**Extended Overflow - Recs**
The number of records in the Extended Overflow area of the file, as of the last time the file was closed.

**Requested Overflow - Recs**
This is the overridden value for the size of the overflow area, in records, when the file was defined or loaded. This value, if provided, will be used when calculating the percentage of overflow used.

**Extended Overflow Blocks**
The number of blocks assigned to the Extended Overflow area.

**Extended Overflow Used**
Indicates, as a percentage, the amount of overflow space used. If a value for overflow records is provided as an override (O=), then the used percentage is based on that value. Otherwise, this is the percentage of the currently allocated extended blocks that are being used.

**Extended PE**
The number of Extended blocks assigned as Prime Extension (PE) blocks.

**Extended Allocated**
The total number of extended area blocks that the file can contain within the currently allocated extents.

**Extended Available**
The number of extended area blocks that are available for use, which could be assigned to either Extended Overflow, or Extended PE.
<table>
<thead>
<tr>
<th><strong>Right Column:</strong></th>
<th><strong>Description of Field Contents:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>EADSCB</td>
<td>Specifies whether or not this dataset has the type of dscb (VTOC entry) that would make the dataset eligible for an EAV volume. Only printed if EATTR has been specified.</td>
</tr>
<tr>
<td>Space Used</td>
<td>The amount of DASD space currently required for the file. (Does not include space required for Extended Available blocks.)</td>
</tr>
<tr>
<td>Space Allocated</td>
<td>For each volume, indicates the amount of DASD space allocated, in tracks.</td>
</tr>
<tr>
<td>Total Space Allocated</td>
<td>Total DASD space allocated, for all volumes.</td>
</tr>
<tr>
<td>Secondary Space</td>
<td>Indicates the amount of space to be requested when a secondary extent is required, as specified on the file defines.</td>
</tr>
<tr>
<td>Max Secondary</td>
<td>The maximum amount of DASD space IAM will request, when additional DASD space is needed for this dataset.</td>
</tr>
<tr>
<td>Share Options</td>
<td>Indicates the defined cross-region share option.</td>
</tr>
<tr>
<td>Index Compress</td>
<td>Indicates whether a compressed index structure exists for this file.</td>
</tr>
<tr>
<td>Dictionary Saved</td>
<td>When the IAM file is compressed by hardware, indicates either YES that the dictionary has been saved within the dataset, or NO that the dictionary is not contained within the dataset.</td>
</tr>
<tr>
<td>RLS USAGE</td>
<td>Indicates if IAM RLS processing is REQUIRED or only REQUESTED for this dataset. This will only be displayed if the IAM CREATE Override of RLSID was specified when this dataset was defined or loaded. The value displayed of REQUESTED or REQUIRED is based on the settings of the RLS keyword in the IAM Global Options Table. If OPTIONAL is specified, then the value will be REQUESTED, otherwise it will indicate REQUIRED.</td>
</tr>
<tr>
<td>Unique Keys</td>
<td>For an alternate index, indicates if the alternate keys are UNIQUE (YES) or NONUNIQUE(NO). Unique keys means that each alternate key can only index one data record in the base cluster. A value of NO means non-unique keys, where each alternate key can index multiple data records in the base cluster.</td>
</tr>
<tr>
<td>Inserts</td>
<td>Indicates the number of records added since the file was last loaded.</td>
</tr>
<tr>
<td>Deletes</td>
<td>Indicates the number of records deleted since the file was last loaded.</td>
</tr>
<tr>
<td>High Allocated RBA</td>
<td>Total number of bytes of DASD storage allocated to the file. For ESDS type files, this value can actually be lower than the High Used RBA field. The reason is that the High Used is based on the length of the actual user data. If the file is data compressed, the high allocated could be substantially lower.</td>
</tr>
<tr>
<td>Alternate Key Size</td>
<td>Field is printed for a Path type of dataset only, indicates the size of the key for the data accessed via this path. For a path on an alternate index, this will be the alternate key size. For a path on a base cluster, this will be the primary key size.</td>
</tr>
<tr>
<td>Alternate Key Offset</td>
<td>Field is printed for a Path type of dataset only. This is the key offset for the data as accessed via this path. For a path on an alternate index, this is the offset of the alternate key within the prime record. For a path on a base cluster, this is the offset of the primary key.</td>
</tr>
<tr>
<td>Journal Records</td>
<td>Indicates what type of journal records will be produced. Possible values include ALL, BEFORE, AFTER or NONE.</td>
</tr>
</tbody>
</table>
There are a few additional informational messages that may appear based on the status of the file at the time the LISTCAT was done. Unless they do not reflect the actual file status, they generally are of no major concern. These messages include:

- FILE HAS NOREUSE ATTRIBUTE, CAN NOT BE RELOADED
- FILE REORG HAS STARTED, AND HAS NOT COMPLETED
- FILE REORGANIZATION IS RECOMMENDED
- FILE IS OPEN, STATISTICS MAY BE INACCURATE

Associations

For IAM datasets that are part of an Alternate Index Sphere, there will be an Associations section of the report. This section will follow all of the above described sections, and include an entry for each associated dataset. The dataset name will be preceded by the type of dataset (Cluster, AIX, or Path). Then after each AIX entry name, the UPGRADE or NOUPGRADE attribute will be displayed. For Path’s, the UPDATE or NOUPDATE attribute will be displayed. The dataset names displayed are those that have been defined. If one or more of the associated datasets has been renamed, copied to a new name, or restored with a new name, then a DEFINE RECATALOG must be performed to update the association information. See Section 10.65 for information on performing a Define Recatalog.
IAMINFO Report

IAM will provide a one page run time report on dataset usage, which is produced each time an IAM dataset is closed providing that the job has an IAMINFO DD card. IAM will also optionally produce an SMF record with the same information, if so enabled by the IAM Global Options table. These reports contain information on the dataset attributes, various statistics about the content of the dataset, statistics on the resource usage, and statistics on the different types of I/O requests processed. The IAM Overrides used for each dataset are also included in the report, adjacent to the statistic that would be most influenced by the override. The IAMINFO reports are the primary information source for what is going on with each dataset. Many of the questions about the resource usage or performance on any particular IAM dataset can be answered with the IAMINFO reports.

The IAMINFO reports are easily obtained by providing an IAMINFO DD card in each job step that processes an IAM dataset. Normally this DD statement indicates a SYSOUT file, but can indicate an actual sequential dataset if desired. The only additional overhead when using IAMINFO is the overhead of actually formatting and writing out the report. IAM always maintains the various statistics that are reported on with IAMINFO, regardless of whether or not the DD statement is in the JCL.

IAM can also optionally produce SMF records that contain all the information that appears on the IAMINFO reports. To enable this facility, the IAM Global Option SMF=YES must be specified, along with indicating the SMF record type to use by specifying RECTYPE=nnn. IAM provides the IAMSMF program that can produce IAMINFO reports from the SMF data. There is also the IAMSMFVS program, which produces more of a summarization set of reports from the IAM SMF data. Additionally, customers can provide their own programs to report on the IAM SMF data.

As can be seen from the example below, all that needs to be done to obtain IAMINFO reports for job steps that process IAM datasets is to add an IAMINFO DD card.

```
//PROCESS EXEC PGM=anypgm
//SYSPRINT DD   SYSOUT=*?
//iamfile DD   DISP=SHR,DSN=my.iam.dataset
//IAMINFO DD   SYSOUT=*? added DD card
```

Figure 10-57: Example of adding an IAMINFO DD card (EX1072A)
IAMINFO for File Load

There are two basic IAMINFO reports. The first is for a file load, for which a sample is provided below. The file load is indicated in the IAM361 message, where it indicates that the dataset was opened for FILE CREATION. Amongst the unique features of the IAMINFO file load report are that the average and maximum record lengths are reported on. These lengths are recorded after data compression, if any. Also reported on, under the IAM365 message, is the amount of data space used as a temporary work area for the index. If the DISK BLOCKS READ is not zero, then a temporary dataset was used for the work area for the index, instead of a Data Space.

Figure 10-58: Sample of an IAMINFO Report for a File Load
IAMINFO for File Access

A sample IAMINFO report for normal file access is shown below. A file access is indicated on the IAM361 message, where it will indicate either that the file is opened for INPUT or UPDATE processing. Other differences from the file load report include statistics for IAM's Dynamic Tabling feature, and statistics for the Index Space, which is a Data Space used to hold the index for the dataset.

![IAMINFO Report](image)

---

**IAM360**

**STEP** - LRG102D **DDNAME** - VSAMCRT1 **DATA SET MONITORED** - IAMV.LRG102.CLUSTER


**IAM362**

**IAM DATA CHARACTERISTICS** -

- **LARGE-FRMT** - IAM FILE STATUS=
  - Loaded
- **LOGICAL RECORD LENGTH**=
  - 1360
- **CI SIZE**=
  - 8192
- **KEY SIZE**=
  - 12
- **TOTAL RECORDS**=
  - 8000000
- **TOTAL RECORDS DELETED**=
  - 0
- **TOTAL RECORDS UPDATED**=
  - 0
- **TOTAL RECORDS INSERTED**=
  - 4000000

**IAM363**

**IAM FILE CHARACTERISTICS** -

- **BLOCKING FACTOR**=
  - 4
- **BLOCK SIZE**=
  - 13682
- **TRACKS IN USE**=
  - 200324
- **VARIABLE LENGTH OVERFLOW**=
  - Yes
- **NUMBER OF IAM DATA BLOCKS**=
  - 500002
- **HIGHEST ALLOCATED RBN**=
  - 1040699
- **FILE DEFINED DATE**=
  - 2014.294
- **FILE DEFINED TIME**=
  - 14:41:04
- **FILE LOADED DATE**=
  - 2014.294
- **FILE LOADED TIME**=
  - 14:41:07
- **FILE UPDATE DATE**=
  - 2014.294
- **FILE UPDATE TIME**=
  - 14:47:07

**IAM372**

**IAM EXTENDED AREA CHARACTERISTICS** -

- **EXT. OVERFLOW RECORDS**=
  - 3000000
- **EXT. OVERFLOW BLOCKS**=
  - 300000
- **EXTENDED BLOCKS ALLOCATED**=
  - 540045
- **EXTENDED BLOCKS USED**=
  - 300639
- **EXTENDED BLOCKS AVAILABLE**=
  - 239406

**IAM373**

REORGANIZATION OF THIS FILE IS RECOMMENDED

**IAM365**

**IAM EXECUTION STATISTICS** -

- **TOTAL STORAGE REQUIRED**=
  - 8450048
- **PRIME INDEX(COMPRESSED)**=
  - 2860056
- **STORAGE ABOVE THE LINE**=
  - 8441856
- **COMPRESSED DATA STRUCTURE**=
  - No
- **64-BIT BUFFER STORAGE (K)**=
  - 74240
- **TOTAL JOB 64-BIT INDEX (K)**=
  - 74240
- **REQUESTS PROCESSED**=
  - 8000004
- **REQUESTS FAILED**=
  - 1
- **DISK BLOCKS READ**=
  - 318300
- **DISK BLOCKS WRITTEN**=
  - 0
- **DYNAMIC BUFFER RETRIEVALS**=
  - 6482555
- **MAXIMUM BUFFERS USED**=
  - 512
- **MAXBUFNO=512
  - **DYNAMIC TABLE RETRIEVALS**=
  - 0
- **Z/HPF I/O REQUESTS**=
  - 318300
- **ECKD I/O REQUESTS**=
  - 0

**IAM368**

SPECIFYING A BUFNO VALUE GREATER THAN 512 MAY IMPROVE PERFORMANCE

**IAM366**

**IAM COMMAND EXECUTION SUMMARY** -

- **GET RANDOM**=
  - 0
- **PUT UPDATE**=
  - 0
- **GET SEQUENTIAL**=
  - 8000001
- **PUT ADD**=
  - 0
- **GET PREVIOUS**=
  - 0
- **POINT (START BROWSE)**=
  - 1
- **GET KGE/GENERIC**=
  - 0
- **POINT KGE/GENERIC**=
  - 0
- **GET (SKIP SEQUENTIAL)**=
  - 0
- **ERASE**=
  - 0
- **ENDREQ**=
  - 0
- **IAM FLUSH BUFFER**=
  - 0
- **CLOSE**=
  - 1
- **OPEN**=
  - 1
- **VERIFY**=
  - 0
- **INVALID REQUESTS**=
  - 0
- **SEQ CHAINED BLOCKS READ**=
  - 782560
- **SEQ CHAINED BLOCKS WRITTEN**=
  - 0

---

Figure 10-59: Sample IAMINFO Report for File Access
<table>
<thead>
<tr>
<th>IAMINFO Report Description</th>
<th>The following tables will describe the various fields that appear on the IAMINFO Report. The first header line, IAM400 indicates the version and level of IAM that processed the dataset, and the date of the report. The second header line, IAM360, provides the Step Name, the DD Name of the file being processed, and the Dataset Name. The third header line, IAM361, indicates the name of the program that opened the IAM dataset, file processing mode (i.e. Creation, Input, or Update), and the open and close time stamps.</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAM362 Data Characteristics</td>
<td></td>
</tr>
<tr>
<td>IAM FILE FORMAT</td>
<td>Indicates if the file is an ENHANCED format IAM dataset, which may also be indicated as LARGE-FRMT if it is in a Large Sequential format dataset (z/OS 1.7 and above, or SMS-EXT for a SMS Extended format. The other possibility COMPATIBLE format IAM dataset, which is required for use by the older non-vsam interfaces.</td>
</tr>
<tr>
<td>IAM FILE STATUS</td>
<td>Indicates whether the file is in a LOADED state, or an UNLOADED state.</td>
</tr>
<tr>
<td>LOGICAL RECORD LENGTH</td>
<td>The maximum record length, as specified when the dataset was defined.</td>
</tr>
<tr>
<td>CI SIZE</td>
<td>The control interval (CI) size specified when file was defined. If no explicit CI size was provided, then IAM calculates a value based on the maximum record size.</td>
</tr>
<tr>
<td>KEY SIZE</td>
<td>The length of the key specified when file was defined. The maximum key length for an IAM dataset is 249 bytes.</td>
</tr>
<tr>
<td>KEY OFFSET</td>
<td>The relative position of the key within the user data record. (RKP) The key must be positioned within the first 4K of the data record.</td>
</tr>
<tr>
<td>AVERAGE RECORD LENGTH</td>
<td>During a load operation IAM calculates an average length for the records that were loaded into the file. For data compressed files, the length is after compression. NOTE: This field only appears for a file CREATION report.</td>
</tr>
<tr>
<td>LARGEST RECORD PROCESSED</td>
<td>During a load operation IAM reports the length of the largest record that was written to the file. NOTE: This field only appears for a file CREATION report.</td>
</tr>
<tr>
<td>TOTAL RECORDS</td>
<td>The total number of records in the IAM file. NOTE: This field does not appear on a file CREATION report.</td>
</tr>
<tr>
<td>TOTAL RECORDS DELETED</td>
<td>The total number of records deleted from the IAM file since creation. NOTE: This field does not appear on a file CREATION report.</td>
</tr>
<tr>
<td>TOTAL RECORDS UPDATED</td>
<td>The total number of records updated in the IAM file since creation. NOTE: This field does not appear on a file CREATION report.</td>
</tr>
<tr>
<td>TOTAL RECORDS INSERTED</td>
<td>The total number of records inserted into the IAM file since creation. NOTE: This field does not appear on a file CREATION report.</td>
</tr>
</tbody>
</table>
IAMINFO Report 10.72

IAM363  File Characteristics

**BLOCKING FACTOR**
A value of 1 to 15 is the number of blocks per track (as blocking factor). A larger value indicates the requested block size from the IAM CREATE B= override.

**BLOCKSIZE**
Physical block size of the IAM file. This is the actual block size developed and used by IAM. NOTE: IAM will determine the optimal block size for a file based upon the file's record length, the blocking factor requested and the track capacity of the device. The IAM block size is transparent to application programs even when the block size is altered or the file is relocated to a different device type.

**TRACKS IN USE**
Number of tracks currently being used by IAM for the file. Does not include unused (available) Extended blocks.

**VARIABLE LENGTH OVERFLOW**
A YES value indicates that IAM will fit as many records as possible into an extended overflow block. A NO value indicates that IAM will only put in the number of maximum size records that will fit within an overflow block. A value of PRO indicates that the Prime Related Overflow structure is being used for this dataset.

**NUMBER OF EXTENTS**
The total number of DASD extents allocated to the IAM file.

**NUMBER OF VOLUMES**
The number of volumes allocated to the IAM file. This may include volumes that do not have any actual space allocated to the dataset, but will not include DFSMS candidate volumes.

**DATASET TYPE**
Describes the type of dataset IAM is simulating. Possible values include: KSDS, for key sequenced dataset, ESDS for entry sequence, (i.e. sequential), ESDS/P for ESDS files defined with PSEUDORBA specified, AIX for an alternate index, or RRDS for an RRDS dataset.

**SHARE OPTIONS**
Indicates the defined cross region share option for this file. Possible values are: 1, 2, 3, or 4. The use of IAM files with Share Option 3 or 4 is not recommended, as possible data loss can occur.

**NUMBER OF IAM DATA BLOCKS**
The number of blocks in the file up to where the prime index begins. This field may be needed to run IAMRECVR for recovery if the first block in the file has been damaged.

**HIGH ALLOCATED RBN**
This the highest allocated block number currently in the IAM file. This field may be needed to run IAMRECVR for recovery if the control information about the file has been damaged.

**INTEGRATED OVERFLOW (CI%)**
The amount of space, as a percentage, left free in each prime data as the file is being loaded. The space can subsequently be used for file expansion. This value is specified as and is similar in concept to CI Freespace.

**DASD RESERVE (CA%)**
Indicates the CA Freespace value that was specified when the file was defined. This is used by IAM to reserve some space for future expansion during automatic space release, which occurs during the first file load.

**FILE DEFINED DATE / TIME**
The date and time that the file was defined.

**FILE LOADED DATE / TIME**
The date and time of the completion of the last file load or reorganization for this file.

**FILE UPDATE DATE / TIME**
The date and time the file was last closed by a program that updated the file.
IAM372 Extended Area Characteristics

EXT. OVERFLOW RECORDS
Number of records currently contained in extended overflow blocks.

EXT. OVERFLOW BLOCKS
Number of formatted extended overflow blocks.

EXTENDED BLOCKS ALLOCATED
Number of extended overflow blocks that will fit within the current amount of DASD space allocated to the file.

EXTENDED PE BLOCKS
Number of extended blocks assigned as Prime Extension (PE) blocks.

EXTENDED BLOCKS USED
Total number of extended overflow blocks in use. This includes Overflow, PE, and Extended Index blocks.

EXTENDED BLOCKS AVAILABLE
The number of extended blocks that are available for use within the currently allocated DASD space.

IAM365 Execution Statistics

TOTAL STORAGE REQUIRED
Total virtual storage from the job's address space acquired by IAM during this execution for this dataset used for control blocks, index areas, and buffers. This includes both above the 16MB line and below the 16MB line, but does not include any 64-bit storage used.

PRIME INDEX [COMPRESSED]
Indicates the amount of virtual storage required for the index to the prime data area of this file. Also, if the index is in a compressed format, the word COMPRESSED appears.

STORAGE ABOVE THE LINE
Total amount of virtual storage acquired above the 16MB line.

COMPRESSED DATA STRUCTURE
Indicates whether data compression was used for this file.

64-BIT BUFFER STORAGE (K)
The amount, in megabytes, of 64-bit virtual storage used for buffers.

TURBO BUFFERING
Indicates whether or not the IAM Real Time Tuning is enabled for Turbo mode, which provides for improved responsiveness when a dataset has heavy physical I/O occurring.

INDEX SPACE USED (K) or 64-BIT INDEX STORAGE(K)
The amount of virtual storage, in KB, used by this dataset for the index when the index is in either a z/OS Data Space or in 64-bit addressable storage.

TOTAL JOB INDEX SPACE USED or 64-BIT INDEX(K)
The total amount of virtual storage, in KB, used so far by IAM in this job step in either a z/OS Data Space or in 64-bit addressable that was acquired for the IAM Index Structure.

REQUESTS PROCESSED
Number of requests made against the file, since last OPEN, by the application (a breakdown by command type follows in the command execution summary).

REQUESTS FAILED
Number of requests made to IAM which did not complete normally (i.e. EOF on sequential read, no record found on random read)
<table>
<thead>
<tr>
<th><strong>DISK BLOCKS READ</strong></th>
<th>Number of physical I/O's used to read blocks from the file.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DISK BLOCKS WRITTEN</strong></td>
<td>Number of physical I/O's used to write blocks to the file, except during a load of a SMS Extended format dataset, in which case this is will be the number of blocks written to disk.</td>
</tr>
<tr>
<td><strong>DYNAMIC BUFFER RETRIEVALS</strong></td>
<td>Number of times that IAM was able to retrieve a block from the buffer pool without the need for a physical I/O.</td>
</tr>
<tr>
<td><strong>MAXIMUM BUFFERS USED</strong></td>
<td>The maximum number of data buffers acquired by IAM's Real Time Tuning during this execution. This may be accompanied by an IAM367 or IAM368 informational message indicating that if IAM was allowed to acquire more buffers the number of I/O's required to service the requests against the file could have been reduced. All buffers are acquired above the 16MB line.</td>
</tr>
<tr>
<td><strong>DYNAMIC TABLE RETRIEVALS</strong></td>
<td>Number of random read requests satisfied by IAM's Dynamic Tabling or by IAM's Dynamic Data Space. Those features enable the caching of records in virtual storage. IAM's Dynamic Table (DYNCORE) is maintained in virtual storage above the 16MB line. The Dynamic Data Space (DYNDS) is a separate Data Space containing the records.</td>
</tr>
<tr>
<td><strong>DYNAMIC TABLE RECORDS</strong></td>
<td>The total number of data records IAM placed into the Dynamic Table or the Dynamic Data Space.</td>
</tr>
<tr>
<td><strong>DATA SPACE USED(M BYTES)</strong></td>
<td>The amount of area actually used in the data space during file load.</td>
</tr>
<tr>
<td><strong>DATA SPACE SIZE(M BYTES)</strong></td>
<td>For a file load, the size of the data space created during file load.</td>
</tr>
<tr>
<td><strong>Z/HPF I/O REQUESTS</strong></td>
<td>The number of physical I/Os that used z/HPF architecture I/O</td>
</tr>
<tr>
<td><strong>ECKD I/O REQUESTS</strong></td>
<td>The number of physical I/Os that used the standard ECKD or CKD type channel program</td>
</tr>
</tbody>
</table>
### IAM366 Command Execution Summary

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET RANDOM</td>
<td>Number of random READ requests with an exact key specified.</td>
</tr>
<tr>
<td>PUT UPDATE</td>
<td>Number of UPDATE requests.</td>
</tr>
<tr>
<td>GET SEQUENTIAL</td>
<td>Number of sequential GET requests.</td>
</tr>
<tr>
<td>PUT ADD</td>
<td>Number of INSERT requests.</td>
</tr>
<tr>
<td>GET PREVIOUS</td>
<td>Number of GET PREVIOUS requests.</td>
</tr>
<tr>
<td>POINT (START BROWSE)</td>
<td>Number of POINT requests.</td>
</tr>
<tr>
<td>GET CI (SEQUENTIAL)</td>
<td>Number of sequential control interval GETS. (ESDS only)</td>
</tr>
<tr>
<td>PUT CI (UPDATE)</td>
<td>Number of CONTROL INTERVAL UPDATES (ESDS only)</td>
</tr>
<tr>
<td>GET CI (RANDOM)</td>
<td>Number of CONTROL INTERVAL GETS (ESDS only)</td>
</tr>
<tr>
<td>PUT CI (ADD)</td>
<td>Number of CONTROL INTERVAL ADDS (ESDS only)</td>
</tr>
<tr>
<td>GET KGE/Generic</td>
<td>Number of random or skip sequential reads, indicating search for key greater or equal, and/or partial key search. (KSDS only)</td>
</tr>
<tr>
<td>POINT KGE/Generic</td>
<td>Number of POINT, or START BROWSE, requests indicating a search for key greater or equal, or partial key search. (KSDS only)</td>
</tr>
<tr>
<td>GET (SKIP SEQUENTIAL)</td>
<td>Number of GET's issued in Skip Sequential mode, with an exact key specified. (KSDS only)</td>
</tr>
<tr>
<td>ERASE</td>
<td>Number of requests to delete records. (KSDS only)</td>
</tr>
<tr>
<td>ENDREQ</td>
<td>Number of LSR ENDREQ requests</td>
</tr>
<tr>
<td>WRTBFR</td>
<td>Number of LSR WRTBFR requests</td>
</tr>
<tr>
<td>IAM STATISTICS</td>
<td>Number of IAM statistic requests</td>
</tr>
<tr>
<td>IAM FLUSH BUFFER</td>
<td>Number of IAM flush buffer requests</td>
</tr>
<tr>
<td>CLOSE</td>
<td>Number of CLOSE requests</td>
</tr>
<tr>
<td>OPEN</td>
<td>Number of OPEN requests</td>
</tr>
<tr>
<td>CLOSE TYPE=T</td>
<td>Number of TEMPORARY CLOSE requests</td>
</tr>
<tr>
<td>VERIFY</td>
<td>Number of VERIFYs</td>
</tr>
<tr>
<td>INVALID REQUESTS</td>
<td>Number of requests with RPL error's</td>
</tr>
<tr>
<td>RECORD LENGTH CHANGES</td>
<td>Number of records with length changes</td>
</tr>
<tr>
<td>SEQ CHAINED BLOCKS READ</td>
<td>Number of chained buffers during READ I/O.</td>
</tr>
<tr>
<td>SEQ CHAIN WRITTEN</td>
<td>Number of chained buffers during WRITE I/O.</td>
</tr>
</tbody>
</table>
As indicated previously, if the IAM Global Options have been set to enable the recording of IAM SMF records, then the IAMSMF program can be used to print IAMINFO reports from the SMF data. This is a handy way to make sure that the information will be available, without having to put IAMINFO DD cards in all of the jobs using IAM datasets. IAMSMF can be run either against the live SMF dataset, or SMF history datasets. To obtain the reports, use the IAMINFO command of IAMSMF.

The IAMINFO command offers a variety of selection criteria, so that you can limit the number of reports produced, if desired. Amongst the criteria that can be specified are job names, dataset names, by dates, or by various dataset attributes or activity levels. Full details on the various options and keywords are available in the System Analysis Utilities Section 41.04 of the manual.

An example of running the IAMSMF utility to print IAMINFO reports is shown below. The DD cards required are SYSPRINT for the printed report(s), SYSIN for the control card input, and SYSMF which is the dataset containing the SMF data. In the example, IAMSMF will print out IAMINFO reports only for the specified job name, plus MERGE=NO means that a report is produced for each IAM dataset close.

```
//IAMINFO EXEC PGM=IAMSMF,REGION=0M
//SYSPRINT DD   SYSOUT=* 
//SYSMF    DD   DISP=OLD,DSN=my.smf.data
//IAMCSV DD DSN=my.iaminfo.csv,DISP=(,CATLG),
// SPACE=(CYL,(5,1)),STORCLAS=storclas
//SYSIN    DD   *

IAMINFO JOBNAME=myjob,MERGE=NO
```

*Figure 10-60: Example JCL for IAMINFO Reports from IAMSMF (EX1072B)*

The IAMINFO reports that are produced by IAMSMF have the IAM360 and IAM361 messages replaced by an IAM370 Job Characteristics Section. The fields in that section are described below.

**IAM370 Job Characteristics**

- **JOB NAME**: Indicates the name of the job that processed the IAM dataset.
- **STEP NAME**: Indicates the job step name that processed the IAM dataset.
- **PROGRAM NAME**: The name of the program that opened the IAM dataset.
- **FUNCTION**: Indicates the function, which can be File Creation, Input Processing, or Update Processing.
- **DDNAME**: The DD name that was processed.
- **DSNAME**: The name of the IAM dataset processed.
- **DATE OPENED**: The date that the dataset was opened.
- **TIME OPENED**: The time that the dataset was opened.
- **DATE CLOSED**: The date that the dataset was closed.
- **TIME CLOSED**: The time that the dataset was closed.
Another reporting alternative, if the IAM SMF records are being recorded, is to use the IAMSMFVS program. IAMSMFVS provides a handy way to monitor and track the use of IAM within an installation, or by particular applications. This program produces three reports, an EXCP report, a Dataset Report, and a Size Report. Each report summarizes activity with one line per dataset and reports on different statistics and attributes. The EXCP report and the Size report by default report on the top 100 datasets, while the Dataset Report contains all of the datasets that were processed. Some customers have used the IAMSMFVS report to watch and determine when various IAM datasets may need to be reorganized based on percentage of the overflow area in use.

An example of the JCL and control card to run IAMSMFVS is shown below. Full details on running IAMSMFVS and on the control card input are provided in the System Analysis Utilities Section 40 of the manual. Please note that IAMSMFVS requires SMF type 30, subtype 4 records or SMF type 4 records to produce reports. There is one keyword specified on the example REPORT control card, which is DSORG=IAM. By default, IAMSMFVS will produce reports on IAM and VSAM. (Note: For VSAM only, specify DSORG=AM).

```plaintext
//IAMSMFVS EXEC PGM=IAMSMFVS,REGION=0M
//SYSMF DD   DISP=SHR,DSN=my.smf.data
//SYSPRINT DD   SYSOUT=*
//SORTIN DD  UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTOUT DD  UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK01 DD  UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK02 DD  UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK03 DD  UNIT=SYSDA,SPACE=(CYL,(50,10))
//SYSOUT DD   SYSOUT=*
//SYIN DD *
   REPORT DSORG=IAM
/*
```

A few additional keywords for the REPORT control card that you might find useful for IAM reporting include:

**CURRENT** - Specifies that the current overflow usage values be reported on, rather than the maximum amount encountered in the data.

**DETAIL** - The EXCP report will contain a breakdown for each use of the file, including job, step, and program name information.

**DSG** - Specifies only datasets that begin with the value(s) specified will be included in the reports. For example, DSG=PROD, or DSG=(PROD1,PROD2.AP,PROD3.AR)

**MAXDSNS** - Specifies the maximum number of datasets that IAMSMFVS will accumulate statistics for. The default value is 60000. Note that for VSAM, each component counts as a dataset, plus the cluster name. So, a KSDS will require 3 entries, and an ESDS will require 2 entries. IAM datasets only require 1 entry each.
**MAXREPORTS** - Specifies the number of datasets that will be included in the EXCP and the SIZE reports. Default is 100.

The first page of the IAMSMFVS output consists of a summarization of all of the data that was selected for reporting, including totals for dataset names, job names, total EXCP counts by access method, and DASD space used by access method (i.e., IAM and VSAM). Below is sample output of the first page.

```
IAM400 SMF DATASET REPORT PROGRAM-IAMSMFVS VER 9.2/00P PIN LEVEL 00-INNOVATION DATA PROCESSING DATE - 2014.295
IAM303 CARD IMAGE - * REPORT DSORG=IAM 00130009*
IAM491 SMF REPORT FUNCTION STARTED - 10.59.29
IAM601 SMF RECORDS -- READ....132651 USED.....13912 DROPPED........0
SPACE UTILIZATION SUMMARY -
  DEVICE TYPE.......3390   VSAM CYLINDERS..........0 IAM CYLINDERS.......78642
  TOTAL DISK EXCPS........7941755 VSAM EXCPS.................0 IAM EXCPS...........4284864
```

*Figure 10-62: Sample IAMSMFVS Summary Report*
After the Summary Report, IAMSMFVS produces the EXCP report. There is a separate EXCP report for IAM datasets and VSAM datasets, if both types of datasets are being reported on. In our sample case, only IAM datasets are being reported on, so there is only an IAM EXCP report. Below is a sample of an IAM EXCP report, as produced by IAMSMFVS. Entries are sorted by largest to smallest EXCP count. As stated previously, only the top 100 datasets are included, unless MAXREPORTS has been specified with a different value.

### IAM EXCP REPORT

<table>
<thead>
<tr>
<th>DATASET NAME</th>
<th>USE COUNT</th>
<th>TOTAL USE</th>
<th>READS</th>
<th>INSERTS</th>
<th>UPDATES</th>
<th>DELETES</th>
<th>OVERFLOW</th>
<th>USED</th>
<th>TRKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAMV.KSD322.CLUSTER</td>
<td>2</td>
<td>146577</td>
<td>127000</td>
<td>126001</td>
<td>67000</td>
<td>0</td>
<td>67000</td>
<td>91</td>
<td>31771</td>
</tr>
<tr>
<td>IAMV.KSD502.SIGAM.CLUSTER</td>
<td>3</td>
<td>93052</td>
<td>5274000</td>
<td>290000</td>
<td>290000</td>
<td>20000</td>
<td>14263</td>
<td>11411</td>
<td>173273</td>
</tr>
<tr>
<td>IAMV.KSD501.SIGAM.CLUSTER</td>
<td>2</td>
<td>39642</td>
<td>5267116</td>
<td>188281</td>
<td>14263</td>
<td>181398</td>
<td>11411</td>
<td>173276</td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD401.CLUSTER</td>
<td>6</td>
<td>36827</td>
<td>90000</td>
<td>330015</td>
<td>0</td>
<td>60000</td>
<td>91</td>
<td>2252</td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD431.CLUSTER</td>
<td>4</td>
<td>28054</td>
<td>160000</td>
<td>264000</td>
<td>20000</td>
<td>16000</td>
<td>13711</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD433.CLUSTER</td>
<td>4</td>
<td>28054</td>
<td>160000</td>
<td>264000</td>
<td>20000</td>
<td>16000</td>
<td>13711</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD432.CLUSTER</td>
<td>3</td>
<td>26223</td>
<td>160000</td>
<td>264000</td>
<td>20000</td>
<td>16000</td>
<td>13711</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD434.CLUSTER</td>
<td>3</td>
<td>25827</td>
<td>160000</td>
<td>264000</td>
<td>20000</td>
<td>16000</td>
<td>13711</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD435.CLUSTER</td>
<td>4</td>
<td>21417</td>
<td>160000</td>
<td>264000</td>
<td>20000</td>
<td>16000</td>
<td>13711</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD430.CLUSTER</td>
<td>4</td>
<td>20778</td>
<td>160000</td>
<td>264000</td>
<td>20000</td>
<td>16000</td>
<td>13711</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD291.CLUSTER</td>
<td>2</td>
<td>20713</td>
<td>160000</td>
<td>264000</td>
<td>20000</td>
<td>16000</td>
<td>13711</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD425.CLUSTER</td>
<td>5</td>
<td>13023</td>
<td>62000</td>
<td>310005</td>
<td>24000</td>
<td>0</td>
<td>12000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD423.CLUSTER</td>
<td>4</td>
<td>12969</td>
<td>160000</td>
<td>850006</td>
<td>220000</td>
<td>0</td>
<td>109997</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD427.CLUSTER</td>
<td>5</td>
<td>10687</td>
<td>62000</td>
<td>310005</td>
<td>24000</td>
<td>0</td>
<td>109997</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD428.CLUSTER</td>
<td>5</td>
<td>12042</td>
<td>62000</td>
<td>310005</td>
<td>24000</td>
<td>0</td>
<td>109997</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD427.CLUSTER</td>
<td>5</td>
<td>10687</td>
<td>62000</td>
<td>310005</td>
<td>24000</td>
<td>0</td>
<td>109997</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD321.CLUSTER</td>
<td>3</td>
<td>9468</td>
<td>67100</td>
<td>66101</td>
<td>67000</td>
<td>0</td>
<td>2588</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD421.CLUSTER</td>
<td>5</td>
<td>9302</td>
<td>160000</td>
<td>850006</td>
<td>220000</td>
<td>0</td>
<td>109996</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.KSD470.CLUSTER</td>
<td>5</td>
<td>8764</td>
<td>8539</td>
<td>22352</td>
<td>2629</td>
<td>6120</td>
<td>2279</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Use Count is the number of SMF records processed for the dataset. The following fields represent the accumulation of data from all of the records: EXCPS, READS, INSERTS, UPDATES, and DELETES. If the keyword CURRENT is included on the REPORT request, then the Overflow statistics are from the most recent record, otherwise they are the maximum encountered. The RECORDS and TRACKS USED are the maximum encountered.
The next report is the Dataset Summary Report. This report is sorted by dataset name, and includes all of the datasets selected for this execution of IAMSMFVS. The information presented includes basic dataset attribute information, such as record length, key length, relative key position, and the block size or CI size.

**Figure 10-63: Sample IAMSMFVS Dataset Report**

The RECFM values for IAM datasets have the following meanings:

**VE:** Enhanced format, variable length records

**VO:** Enhanced format, variable overflow

**DC:** Data Compressed

Note that for IAM datasets, the average record length is based only on records initially loaded into the dataset, and if data compression was being used, will be based on the record lengths after compression.

<table>
<thead>
<tr>
<th>DATA SET NAME</th>
<th>USE</th>
<th>TOTAL EXCPS</th>
<th>AVG RECL</th>
<th>MAX LRECL</th>
<th>LEN</th>
<th>RKP</th>
<th>CI%</th>
<th>CA%</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAMV.DCV201.CLUSTER</td>
<td>2</td>
<td>38 IAM VO-DC</td>
<td>64</td>
<td>64</td>
<td>12</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV202.CLUSTER</td>
<td>2</td>
<td>328 IAM VO-DC</td>
<td>670</td>
<td>670</td>
<td>8</td>
<td>11476</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV203.CLUSTER</td>
<td>3</td>
<td>400 IAM VO-DC</td>
<td>256</td>
<td>256</td>
<td>8</td>
<td>11476</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV204.CLUSTER</td>
<td>2</td>
<td>428 IAM VO-DC</td>
<td>512</td>
<td>512</td>
<td>252</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV205.CLUSTER</td>
<td>2</td>
<td>857 IAM VO-DC</td>
<td>1024</td>
<td>1024</td>
<td>516</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV206.CLUSTER</td>
<td>2</td>
<td>5 IAM VO-DC</td>
<td>64</td>
<td>64</td>
<td>12</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV206A.CLUSTER</td>
<td>2</td>
<td>5 IAM VO-DC</td>
<td>64</td>
<td>64</td>
<td>12</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV206B.CLUSTER</td>
<td>2</td>
<td>5 IAM VO-DC</td>
<td>64</td>
<td>64</td>
<td>12</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV206C.CLUSTER</td>
<td>2</td>
<td>5 IAM VO-DC</td>
<td>64</td>
<td>64</td>
<td>12</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV206D.CLUSTER</td>
<td>2</td>
<td>5 IAM VO-DC</td>
<td>64</td>
<td>64</td>
<td>12</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV206E.CLUSTER</td>
<td>2</td>
<td>5 IAM VO-DC</td>
<td>64</td>
<td>64</td>
<td>12</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV206F.CLUSTER</td>
<td>2</td>
<td>5 IAM VO-DC</td>
<td>64</td>
<td>64</td>
<td>12</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV207.CLUSTER</td>
<td>2</td>
<td>2377 IAM VO-DC</td>
<td>540</td>
<td>1040</td>
<td>16</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV208A.CLUSTER</td>
<td>3</td>
<td>16 IAM VO-DC</td>
<td>175</td>
<td>175</td>
<td>8</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV208B.CLUSTER</td>
<td>3</td>
<td>172 IAM VO-DC</td>
<td>175</td>
<td>175</td>
<td>8</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV209.CLUSTER</td>
<td>2</td>
<td>3433 IAM VO-DC</td>
<td>1040</td>
<td>1040</td>
<td>24</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV210.CLUSTER</td>
<td>2</td>
<td>179 IAM VO-DC</td>
<td>64</td>
<td>128</td>
<td>12</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV211.CLUSTER</td>
<td>2</td>
<td>1157 IAM VO-DC</td>
<td>64</td>
<td>1039</td>
<td>12</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV212.CLUSTER</td>
<td>1</td>
<td>13 IAM VO-DC</td>
<td>64</td>
<td>1024</td>
<td>12</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV213.CLUSTER</td>
<td>2</td>
<td>104 IAM VO-DC</td>
<td>64</td>
<td>1024</td>
<td>12</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.DCV214.CLUSTER</td>
<td>2</td>
<td>18 IAM VO-DC</td>
<td>2339</td>
<td>2340</td>
<td>4</td>
<td>23476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.KSD210.CLUSTER</td>
<td>2</td>
<td>113 IAM VO-DC</td>
<td>100</td>
<td>1000</td>
<td>4</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.KSD211.CLUSTER</td>
<td>3</td>
<td>258 IAM VO-DC</td>
<td>100</td>
<td>500</td>
<td>4</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.KSD212.CLUSTER</td>
<td>3</td>
<td>182 IAM VO-DC</td>
<td>100</td>
<td>500</td>
<td>4</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.KSD213.CLUSTER</td>
<td>3</td>
<td>327 IAM VO-DC</td>
<td>100</td>
<td>500</td>
<td>4</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.KSD214.CLUSTER</td>
<td>3</td>
<td>4097 IAM VO-DC</td>
<td>100</td>
<td>2100</td>
<td>4</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.KSD215.CLUSTER</td>
<td>3</td>
<td>152 IAM VO-DC</td>
<td>100</td>
<td>1000</td>
<td>4</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.KSD216.CLUSTER</td>
<td>3</td>
<td>290 IAM VO-DC</td>
<td>100</td>
<td>1000</td>
<td>4</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.KSD217.CLUSTER</td>
<td>2</td>
<td>2034 IAM VO-DC</td>
<td>256</td>
<td>256</td>
<td>12</td>
<td>23476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.KSD220.CLUSTER</td>
<td>2</td>
<td>628 IAM VO-DC</td>
<td>100</td>
<td>1000</td>
<td>4</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.KSD225.CLUSTER</td>
<td>7</td>
<td>1489 IAM VO-DC</td>
<td>1147</td>
<td>2040</td>
<td>12</td>
<td>13682</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.KSD230.CLUSTER</td>
<td>2</td>
<td>391 IAM VO-DC</td>
<td>1000</td>
<td>1020</td>
<td>4</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.KSD231.CLUSTER</td>
<td>2</td>
<td>387 IAM VO-DC</td>
<td>1000</td>
<td>1020</td>
<td>4</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.KSD232.CLUSTER</td>
<td>2</td>
<td>240 IAM VO-DC</td>
<td>1000</td>
<td>1120</td>
<td>4</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IAMV.KSD240.CLUSTER</td>
<td>2</td>
<td>693 IAM VO-DC</td>
<td>100</td>
<td>1000</td>
<td>4</td>
<td>11476</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>
The Size report is broken out in a manner similar to the EXCP report. There are separate reports for IAM and VSAM datasets, and the reports consist of the largest 100 datasets, or whatever value was specified on the MAXREPORTS operand. This report contains the most information about the overflow area use, so it would be the one to use for a determination as to whether reorganization is needed.

<table>
<thead>
<tr>
<th>DATA SET NAME</th>
<th>TRACKS</th>
<th>TOTAL USE</th>
<th>TOTAL REC</th>
<th>INDEPENDENT OVERFLOW</th>
<th>PRIME EXT</th>
<th>CI%</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAMV.KSD322.CLUSTER</td>
<td>31771</td>
<td>146577</td>
<td>2</td>
<td>127000</td>
<td>73357</td>
<td>91</td>
</tr>
<tr>
<td>IAMV.KSD513.CLUSTER</td>
<td>26082</td>
<td>23067</td>
<td>2</td>
<td>572710</td>
<td>23122</td>
<td>81</td>
</tr>
<tr>
<td>IAMV.KSD261.CLUSTER</td>
<td>22465</td>
<td>5998</td>
<td>4</td>
<td>1640000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IAMV.KSD321.CLUSTER</td>
<td>16788</td>
<td>9468</td>
<td>3</td>
<td>67100</td>
<td>7438</td>
<td>34</td>
</tr>
<tr>
<td>IAMV.KSD362.CLUSTER</td>
<td>15409</td>
<td>14787</td>
<td>4</td>
<td>614000</td>
<td>32700</td>
<td>91</td>
</tr>
<tr>
<td>IAMV.KSD322.CLUSTER</td>
<td>15008</td>
<td>1002</td>
<td>1</td>
<td>60000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IAMV.KSD360.CLUSTER</td>
<td>13912</td>
<td>14771</td>
<td>4</td>
<td>554000</td>
<td>49400</td>
<td>90</td>
</tr>
<tr>
<td>IAMV.KSD361.CLUSTER</td>
<td>13912</td>
<td>14692</td>
<td>4</td>
<td>554000</td>
<td>44580</td>
<td>80</td>
</tr>
<tr>
<td>IAMV.KSD502.$IAM.CLUSTER</td>
<td>13711</td>
<td>93052</td>
<td>3</td>
<td>5274000</td>
<td>864552</td>
<td>20</td>
</tr>
<tr>
<td>IAMV.KSD501.$IAM.CLUSTER</td>
<td>13696</td>
<td>39642</td>
<td>2</td>
<td>5267116</td>
<td>864552</td>
<td>20</td>
</tr>
<tr>
<td>IAMV.KSD500.$IAM.CLUSTER</td>
<td>13696</td>
<td>39654</td>
<td>2</td>
<td>5267116</td>
<td>864552</td>
<td>20</td>
</tr>
<tr>
<td>IAMV.KSD251.CLUSTER</td>
<td>5339</td>
<td>716</td>
<td>2</td>
<td>330000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IAMV.KSD433.CLUSTER</td>
<td>1586</td>
<td>28054</td>
<td>4</td>
<td>160000</td>
<td>76637</td>
<td>109991</td>
</tr>
<tr>
<td>IAMV.KSD432.CLUSTER</td>
<td>1586</td>
<td>26223</td>
<td>3</td>
<td>160000</td>
<td>76637</td>
<td>109991</td>
</tr>
<tr>
<td>IAMV.KSD431.CLUSTER</td>
<td>1586</td>
<td>28054</td>
<td>4</td>
<td>160000</td>
<td>76637</td>
<td>109991</td>
</tr>
<tr>
<td>IAMV.KSD434.CLUSTER</td>
<td>1585</td>
<td>25827</td>
<td>3</td>
<td>160000</td>
<td>66110</td>
<td>109991</td>
</tr>
<tr>
<td>IAMV.KSD425.CLUSTER</td>
<td>1557</td>
<td>13023</td>
<td>5</td>
<td>62000</td>
<td>12130</td>
<td>12000</td>
</tr>
<tr>
<td>IAMV.KSD427.CLUSTER</td>
<td>1421</td>
<td>10687</td>
<td>5</td>
<td>62000</td>
<td>12130</td>
<td>10335</td>
</tr>
<tr>
<td>IAMV.KSD428.CLUSTER</td>
<td>1418</td>
<td>12042</td>
<td>5</td>
<td>62000</td>
<td>12170</td>
<td>9656</td>
</tr>
<tr>
<td>IAMV.KSD430.CLUSTER</td>
<td>1384</td>
<td>20778</td>
<td>4</td>
<td>160000</td>
<td>153274</td>
<td>109991</td>
</tr>
<tr>
<td>IAMV.KSD435.CLUSTER</td>
<td>1204</td>
<td>21417</td>
<td>4</td>
<td>160000</td>
<td>66022</td>
<td>109977</td>
</tr>
<tr>
<td>IAMV.KSD216.CLUSTER</td>
<td>1098</td>
<td>290</td>
<td>3</td>
<td>335000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IAMV.KSD331.CLUSTER</td>
<td>1002</td>
<td>206</td>
<td>2</td>
<td>40000</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

For Enhanced Format files, the Maximum Overflow Records are based on either the user specified number of overflow records, from the CREATE O= override. The O= override value is retained for informational purposes to aid in making decisions about when the dataset should be reorganized. IAM makes no guarantee that there is sufficient DASD space for IAM to actually keep that number of records in the overflow area. Also, the overflow may be able to hold more records than the number specified, so the overflow used percentage can exceed 100.

If no override had been provided, then maximum overflow records is calculated based on the number of extended blocks currently allocated to the dataset.
10.74 Reporting on IAM Datasets with FDREPORT

FDREPORT

Customers who have the ABR product from Innovation Data Processing can use FDREPORT to find and provide information on IAM datasets. FDREPORT is a very powerful and flexible reporting tool that enables the user to customize reports with information about datasets residing on their DASD volumes. FDREPORT can pull information from the VTOC, the catalog, the VVDS, and for IAM datasets, statistical and attribute information from the dataset itself, and merge that information into a single report. Using FDREPORT, you can find out where all of your IAM datasets are, which datasets are IAM datasets, how much space is being used by IAM, does a particular volume or volume group have any IAM datasets, plus many other dataset management questions.

Full information on using FDREPORT is in Section 54 of the FDR manual. It is quite easy to use. First, you specify what datasets you want to report on using the SELECT or XSELECT cards. You can also exclude various datasets with the EXCLUDE or XEXCLUDE control cards. Then, using the REPORT card, you specify what fields you want included in your report. Next, you can provide your own report title using the TITLE card. Then, with the PRINT card, FDREPORT goes to work for you, and very quickly you will have your own customized report.

Shown below is a sample of the JCL and control cards that can be used to find all of the IAM datasets in your installation. The information being requested includes:

1. SPLDSN  The dataset name, split across two lines if it is very long.
2. VOL   The volume containing the dataset.
3. SIZE   The space allocated to the dataset, in tracks.
4. %FREE  The percentage of unused space.
5. NOEXTENT  The number of extents.
6. BLKSIZE  The block size of the dataset.
7. MAXLRECL  The maximum record length.
8. KEYLEN  The key length of the dataset.
9. RKP  The relative key position.
10. RECORDS  The number of records in the dataset.
11. OVERUSED  The number of records in the IAM Overflow Area

```
//KSD972C EXEC PGM=FDREPORT,REGION=OM
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
XSELECT DSORG=IAM
REPORT FIELDS=(SPLDSN,VOL,SIZE,%FREE,NOEXTENT,
BLKSIZE,MAXLRECL,KEYLEN,RKP,RECORDS,OVERUSED)
TITLE LINE='IAM DATASET REPORT'
PRINT ENABLE=IAM,SORT=COMBINE,SORTALLOC=(SORTWORK,CYL),ONLINE
/*
```

Figure 10-65: Sample JCL to run FDREPORT (EX1074A)
The report produced by the JCL and control cards above looks like this:

```plaintext
FDR400 FDRABR CUSTOM REPORTS–FDREPORT VER 5.2/65P INNOVATION DATA PROCESSING DATE-10.069 PAGE- 01
IAM DATASET REPORT

<table>
<thead>
<tr>
<th>DATASET NAME</th>
<th>VOLSER</th>
<th>ALLOC %FR</th>
<th>EXT</th>
<th>BKSIZ</th>
<th>MAXLR</th>
<th>LEN</th>
<th>RKP</th>
<th>RECORDS</th>
<th>OVERUSED</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFM.P.EMBOX</td>
<td>IDPLB5</td>
<td>7500</td>
<td>0</td>
<td>3</td>
<td>32760</td>
<td>32750</td>
<td>55</td>
<td>4</td>
<td>651727</td>
</tr>
<tr>
<td>IAMV.$IAM.GAMA.KSDS2</td>
<td>IDPLB2</td>
<td>15</td>
<td>0</td>
<td>1</td>
<td>13682</td>
<td>500</td>
<td>8</td>
<td>4</td>
<td>500</td>
</tr>
<tr>
<td>IAMV.$IAM.GAMA0008.CLUSTER</td>
<td>IDPLB2</td>
<td>15  86</td>
<td>1</td>
<td>13682</td>
<td>80</td>
<td>8</td>
<td>4</td>
<td>185</td>
<td>0</td>
</tr>
<tr>
<td>IAMV.$IAM.GAMA8.CLUSTER</td>
<td>IDPLB2</td>
<td>15  86</td>
<td>1</td>
<td>13682</td>
<td>80</td>
<td>8</td>
<td>4</td>
<td>185</td>
<td>0</td>
</tr>
<tr>
<td>IAMV.KSD972A.CLUSTER</td>
<td>SCR092</td>
<td>750  79</td>
<td>1</td>
<td>13682</td>
<td>1300</td>
<td>4</td>
<td>12</td>
<td>6000</td>
<td>0</td>
</tr>
<tr>
<td>IAMV.KSD972B.CLUSTER</td>
<td>SCR092</td>
<td>750  86</td>
<td>1</td>
<td>13682</td>
<td>1300</td>
<td>4</td>
<td>12</td>
<td>4000</td>
<td>0</td>
</tr>
<tr>
<td>IAMV.KSD972C.CLUSTER</td>
<td>SCR092</td>
<td>750  69</td>
<td>1</td>
<td>13682</td>
<td>1300</td>
<td>4</td>
<td>12</td>
<td>9000</td>
<td>0</td>
</tr>
<tr>
<td>IAMV.VIT212A.CLUSTER</td>
<td>SCR083</td>
<td>30  46</td>
<td>1</td>
<td>23476</td>
<td>2340</td>
<td>4</td>
<td>12</td>
<td>256</td>
<td>0</td>
</tr>
<tr>
<td>MET1.MAILFILE</td>
<td>SCR083</td>
<td>960  0</td>
<td>12</td>
<td>11476</td>
<td>2260</td>
<td>8</td>
<td>4</td>
<td>42587</td>
<td>0</td>
</tr>
<tr>
<td>SYSP.USTPROD.$USTCAT</td>
<td>IDPLB4</td>
<td>150    76</td>
<td>1</td>
<td>11476</td>
<td>504</td>
<td>18</td>
<td>4</td>
<td>686</td>
<td>0</td>
</tr>
<tr>
<td>SYSP.USTPROD.$USTINFO</td>
<td>IDPLB4</td>
<td>6900  4</td>
<td>5</td>
<td>15476</td>
<td>354</td>
<td>50</td>
<td>4</td>
<td>1557834</td>
<td>0</td>
</tr>
</tbody>
</table>
```

*Figure 10-66: Sample Report from FDREPORT*
Example 2:  FDREPORT Selective Criteria

A more selective example is shown below. For this report request, selection criteria has been added to look only for datasets that begin with the specified character string, as identified by the DSG= parameter, that are on volumes that begin with the character string identified by VOLG=.

```plaintext
//KSD972C EXEC PGM=FDREPORT,REGION=0M
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *
XSELECT DSORG=IAM,DSG=prod.ordent,VOLG=IDPLB
REPORT FIELDS=(DSN,VOL,SIZE,NOEXTENT,IAMINFO,KEYLEN)
TITLE LINE='IAM ORDER ENTRY DATASET REPORT'
PRINT ENABLE=IAM,SORT=COMBINE,SORTALLOC=(SORTWORK,CYL),ONLINE /*
```

Figure 10-67: Example FDREPORT for IAM Datasets (EX1074B)

An example of the output is shown below.

```
<table>
<thead>
<tr>
<th>DATASET NAME</th>
<th>D/S</th>
<th>VOLSER</th>
<th>ALLOC</th>
<th>EXT</th>
<th>ORG</th>
<th>RECFM</th>
<th>MAXLR</th>
<th>LRECL</th>
<th>BKSIZ</th>
<th>CISIZ</th>
<th>LEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROD.ORDENT.MEPFILE</td>
<td>IDPLB5</td>
<td>15</td>
<td>1</td>
<td>IAM</td>
<td>FB</td>
<td>16</td>
<td>16</td>
<td>11472</td>
<td>0</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>PROD.ORDENT.NAMEFILE</td>
<td>IDPLB2</td>
<td>60</td>
<td>1</td>
<td>IAM</td>
<td>FB</td>
<td>99</td>
<td>99</td>
<td>13662</td>
<td>0</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>PROD.ORDENT.PRICE.FILE</td>
<td>IDPLB2</td>
<td>5</td>
<td>1</td>
<td>IAM</td>
<td>FB</td>
<td>30</td>
<td>30</td>
<td>13680</td>
<td>11460</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>PROD.ORDENT.PROD.LEVEL</td>
<td>IDPLB2</td>
<td>77</td>
<td>2</td>
<td>IAM</td>
<td>FB</td>
<td>17</td>
<td>17</td>
<td>13668</td>
<td>11475</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>PROD.ORDENT.FILE</td>
<td>IDPLB2</td>
<td>1026</td>
<td>3</td>
<td>IAM</td>
<td>VB</td>
<td>2260</td>
<td>810</td>
<td>13682</td>
<td>11476</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 10-69: Example 2 of FDREPORT Output
IAMXMON: IAM Execution Monitor for CICS

IAMXMON is a CICS program and transaction that provides real-time statistical information on opened Enhanced format files under CICS. The installation of this program is optional, and an installation must assign a transaction name which you must know to use it. Refer to Section 90.12 for information on installing the IAMXMON program and transaction.

Summary Display

By entering the assigned CICS Transaction Identifier (TRANSID), the user is presented with a list of OPEN Enhanced format IAM files, described by DDNAME and Dataset Name. Also displayed is the total amount of CICS storage being used for each IAM dataset. This display is scrollable using the PF7/PF8 keys.

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Dataset Name</th>
<th>Storage</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSDS0</td>
<td>IAMCICSV.$IAM.RLB9.KSDS0</td>
<td>0000012288</td>
<td></td>
</tr>
<tr>
<td>S KSDS4</td>
<td>IAMCICSV.$IAM.RLB9.KSDS4</td>
<td>0000012288</td>
<td></td>
</tr>
<tr>
<td>KSDS3</td>
<td>IAMCICSV.$IAM.RLB9.KSDS3</td>
<td>0000012288</td>
<td></td>
</tr>
<tr>
<td>KSDS2</td>
<td>IAMCICSV.$IAM.RLB9.KSDS2</td>
<td>0000012288</td>
<td></td>
</tr>
<tr>
<td>KSDS1</td>
<td>IAMCICSV.$IAM.RLB9.KSDS1</td>
<td>0000012288</td>
<td></td>
</tr>
</tbody>
</table>

Figure 10-70: Example of Initial IAMXMON Display

Dataset Activity Display

To view the detail information about any displayed file, select it by entering 'S' next to the desired file. The first display of two is displayed showing:

<table>
<thead>
<tr>
<th>Command</th>
<th>Stg Used/High Used</th>
<th>Index Space</th>
<th>ID:</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>000000008</td>
<td>0000000004</td>
<td></td>
</tr>
<tr>
<td>PUT</td>
<td>000000004</td>
<td>0000000000</td>
<td></td>
</tr>
<tr>
<td>Seq</td>
<td>000000000</td>
<td>0000000000</td>
<td></td>
</tr>
<tr>
<td>KGE</td>
<td>000000000</td>
<td>0000000000</td>
<td></td>
</tr>
<tr>
<td>(skip)</td>
<td>000000000</td>
<td>0000000000</td>
<td></td>
</tr>
<tr>
<td>Verify</td>
<td>000000000</td>
<td>0000000000</td>
<td></td>
</tr>
<tr>
<td>Seq Chain R</td>
<td>0000000000</td>
<td>Seq Chain W</td>
<td>0000000000</td>
</tr>
</tbody>
</table>

Figure 10-71: IAMXMON Dataset Activity Display Page
By hitting the ENTER key, a second display describing the physical file will be displayed. Hitting ENTER will switch back and forth between the two displays. Use PF3/PF15 to return to the open dataset list.

Figure 10-72: IAMXMON Dataset Attributes Display Page
Overview

As with any indexed dataset, IAM datasets require periodic maintenance. This includes such chores as periodic reorganizations, renaming, recataloging, backup and restore, moving, and on rare occasion a recovery may be necessary. This section will explain why and when such chores are necessary, and provide plenty of examples on how to accomplish each chore to meet your processing requirements.
10.81 Reorganizing IAM Datasets

Why Reorganize

As with any indexed dataset type, IAM datasets that are updated will have to be periodically reorganized for optimum performance and DASD space usage. As records are added or updated with record length increases, these records are placed into the extended areas (Extended Overflow / Extended PE) of the IAM dataset. This causes the amount of virtual storage required to index the file to increase. As the overflow index grows, there will also be additional CPU time required to open the dataset, and to search and maintain position within the index. Additionally, for sequential processing, keys in ascending key sequence can end up scattered throughout the extended overflow area causing excessive I/O when reading or updating the dataset. Datasets are still subject to running out of extents or DASD space. Deleted records may leave significant portions of the prime area empty and not reusable because the original prime index is no longer reflective of the data in the file. For these reasons, updated files are going to need periodic reorganization.

When to Reorganize

Many applications that have been using VSAM or IAM datasets already have regularly scheduled dataset reorganizations. Generally, there should be no need to change the schedule. For datasets that are only reorganized when needed, the difficulty is in determining when an IAM dataset should be reorganized. With Compatible format IAM files, it was relatively easy because of the fixed size of the overflow area. With Enhanced format files the ability to acquire extents and now with variable overflow, it is not quite as clear cut. Amongst the criteria to consider are:

1. Amount of virtual storage required for the Extended Overflow Index. This can be estimated by multiplying the number of records in Overflow by the key length plus 4.
2. Percentage of records that are in Extended Overflow. For example, if more than 10% of the records in the dataset are in Extended Overflow, then the dataset should be reorganized.
3. Number of extents being used for single volume dataset.
4. The Extended area of the file exceeding a specified quantity of DASD space.
5. Exceeding specified quantity of records in Extended Overflow.

To aid in determining when a dataset should be reorganized, IAM will produce a warning messages during open or close if certain conditions are detected for which reorganization might be warranted. The messages including the numbers have been changed in Version 9.2. The message numbers are IAMW21, IAMW22, IAMW91, and IAMW92. These messages are meant to bring to attention that reorganization should be considered for the indicated datasets. Certainly some datasets may continue processing for several days or weeks without a reorganization after hitting a threshold condition, but only experience will be able to indicate if that is true. Other datasets, such as small datasets with very high activity, may need to be reorganized more frequently. The IAM thresholds include the Extended Overflow index exceeding sixty four megabytes of storage, using thirteen or more extents for a single volume dataset, or when the overflow area exceeds one thousand cylinders. The Overflow override can be used to monitor how much overflow space is being used, similar to what was done for Compatible format IAM files, except that the limit is no longer the absolute limit. By providing the override, IAM will monitor and inform you via an IAM message that reorganization is recommended because the Extended Overflow area exceeds the number of records specified.
Reorganizing IAM Datasets

Reorganization can be done automatically. Innovation Data Processing offers a software package, FDRREORG, which can automate file reorganizations for IAM, VSAM, and PDS types of datasets. Other alternatives used by some customers include using the IAM SMF reports or the IAM SMF records directly to trigger dataset reorganizations, and the IAMW22 message has also been used. Also the IAMPRINT report produced by a LISTCAT can trigger dataset reorganization.

IAM datasets can be easily reorganized with the FDRREORG product from Innovation Data Processing. FDRREORG can search out all or just the selected IAM datasets, and if they meet the specified criteria, will be automatically reorganized. FDRREORG maximizes buffering for fast reorganizations, can reorganize multiple datasets concurrently, and has a feature to DELETE and DEFINE the datasets that are being reorganized. FDRREORG provides the capability to read from each volume of a multivolume IAM dataset concurrently (the MAXPARALLELBACKUPS= keyword), which can result in faster reorganization times for the very large datasets.

The other common way to reorganize an IAM dataset is with IDCAMS REPRO. With appropriate buffering, IDCAMS REPRO can perform the reorganization quickly. The general technique is to REPRO the data from the IAM dataset into a sequential dataset, then optionally delete and redefine the IAM dataset, followed by an IDCAMS REPRO REUSE from the sequential dataset into the IAM dataset. An alternative with IDCAMS is to REPRO from one IAM dataset into another IAM dataset without going to a sequential dataset. This could save considerable time, providing that there is the DASD space available for both copies of the dataset, and that a sequential backup of the data is not required. Some installations have written programs to read the IAMSMFVS report for IAM datasets or to read the optional IAM SMF records directly, and then build and submit job(s) to reorganize the selected files. The IAMPRINT LISTCAT output has also been used for this purpose.

Some application software packages provide their own dataset reorganization utility. The one thing to watch out for with such utilities is to determine if they are doing a single record load followed by a mass insert. Such a process will result in the data records all being placed within the extended areas of the file, which may adversely impact the performance of the file in subsequent executions. If such is the case, then an additional reorganization by IDCAMS should be done following the application reorganization to ensure optimum performance.

IAM base cluster types of datasets with alternate indexes are marked as non-reusable by IAM, same as with VSAM. The base cluster cannot be reorganized without deleting and redefining the base, and all of the related components except by FDRREORG. This is to help insure that the alternate index files retain their integrity with the base cluster. The IAM alternate index datasets can be reorganized as needed, or they can just be rebuilt. FDRREORG V5.3/50 or higher is required to reorganize IAM Alternate Index datasets, or FDRREORG V5.4/20 if any of the IAM datasets are hardware compressed or in a SMS Extended format.

IAM RRDS datasets (both fixed length record and variable length record) may need to be periodically reorganized. The need for RRDS reorganization is based upon a

Alternate Index

How To Reorganize

RRDS
large percentage of the records being in the Extended Overflow area of the dataset. IDCAMS can be used to reorganize RRDS datasets, however there are some special considerations. The unique nature of RRDS files is that where the "key", that is the relative record number, is not stored within the data record as seen by application programs. If there are empty slots (i.e., unused record numbers) within the dataset, IDCAMS does not retain the record number for a typical REPRO when the data is copied to an intermediate sequential dataset. Therefore, to use IDCAMS to reorganize RRDS files, one must either perform a REPRO from an RRDS dataset directly into an RRDS dataset. With IAM, there is another option which is to use the IAM BACKUPCOMPRESSED option.

If you use the method of performing the REPRO from an RRDS directly into another RRDS, you may also want to use the MODEL parameter when defining the target RRDS to make sure it has the proper attributes. The MODEL works fine for fixed length record RRDS files, however it results in a KSDS for variable length RRDS files, apparently because IDCAMS does not recognize that the dataset being modeled is a variable length record RRDS. (Note that this is true for both VSAM and IAM). To use the MODEL parameter successfully with a variable length record RRDS, the RECORDSIZE parameter and the NUMBERED parameter MUST BE SPECIFIED.

When using the IAM BACKUPCOMPRESSED option with IAM RRDS files, the relative record number will be included in the record provided to IDCAMS. When reloading an RRDS with BACKUPCOMPRESSED specified, IAM will check the input record for the relative record number to be used for each record. When performing such a reorganization, the sequential output file must be allocated as a variable length (RECFM=VB) file, with an LRECL of the record length in the IAM file plus 12 bytes.

See Examples F, G, and H for sample JCL to reorganize IAM RRDS files.

An IAM feature that may help speed up the reorganization of software compressed datasets is to not decompress the data while it is being read, and to accept the IAM software compressed data format on the reload. This feature will eliminate the CPU time spent decompressing and compressing the data, and eliminate I/O transfer time to the output device. If the output is a tape device that offers compression, there normally will not be any savings in tape usage because the data is already compressed. FDRREORG will automatically invoke this facility. For IDCAMS REPRO, the IAM ACCESS and CREATE Override of BACKUPCOMPRESSED must be specified on both the backup and the reload portion of the reorganization. Note that the sequential file record length must be eight bytes larger than the defined maximum record size for KSDS files, twelve bytes for 4-byte RBA ESDS and RRDS, and sixteen bytes for 8-byte RBA ESDS dataset and must have a record format of variable blocked (RECFM=VB).

This facility is not recommended for use when the sequential backup is read by other application programs, as the data is not in a usable form. The compressed backup file can be converted to an uncompressed sequential file by the IAMRECVR DECOMPRESS command. For the adventuresome, IAM provides a callable service that will read and decompress a sequential file with compressed IAM data, and can also write out user data to a sequential file in IAM compressed format. Refer to the Reference Section of this manual for further information on using the callable service. Note that while SYNSCORT can be used to reorganize IAM files, it cannot use the BACKUPCOMPRESSED feature because of the mechanisms it uses to determine the file characteristics.
The first example of reorganizing IAM datasets is with FDRREORG. In this example, all cataloged IAM datasets that have a high level index of MYIAM will be considered for reorganization. The criteria being specified is that if any of the following are true, the dataset will be reorganized:

- The Overflow Index exceeds 4 megabytes.
- More than 10% of the records in the file are in the Extended Overflow area.
- There are more than 1,000,000 records in Overflow.

Defaults for the backup datasets and for the maximum tasks will be used.

```assembly
//REORGIAM EXEC PGM=FDRREORG,REGION=0M
//SYSPRINT DD  SYSOUT=*  
//REORGPRT DD  SYSOUT=*  
//REORGRPT DD  SYSOUT=*  
//IAMINFO DD  SYSOUT=*  
//SYSIN  DD  *  
REORG DSTYPE=IAM
SELECT CATDSN=(MYIAM.**),IFANY,
  OVERFLOWINDEX>4194304,
  PCTTRECO>10,
  ORECS>1000000  
/*
```

*Figure 10-73: Reorganization of Multiple IAM Datasets Selected by FDRREORG (EX1081A)*
Example B: Parallel Volume I/O with FDRREORG

In this example, a single multi-volume IAM dataset is reorganized in parallel mode. This feature allows concurrent reading of the different volumes on which the IAM dataset resides, which can reduce reorganization time depending on the I/O configuration for both the source IAM dataset, and the backup datasets. Note that as required by FDRREORG, the BACKUPINDEX is specified with a ? embedded so that each volume backup has a unique name.

```
//REORGIAM EXEC PGM=FDRREORG,REGION=0M
//SYSPRINT DD   SYSOUT=*  
//REORGPRT DD   SYSOUT=*  
//REORGRPT DD   SYSOUT=* 
//IAMINFO DD   SYSOUT=*  
//SYSin    DD   *  
   REORG MODE=P,MAXP=4,BACKUPI=++BACKUP?
   SELECT CATDSN=(MYIAM.DATASET)
/*
```

Example C: IDCAMS Reorg with Delete and Define

In this example, the IAM dataset is reorganized with an IDCAMS REPRO. IAM Overrides are used to not decompress the data. The sequential dataset is being written to a 3390 type of DASD volume, where ½ track blocking is being specified. Note that a permanent dataset name is being given for the sequential file, and that the dataset is being kept and cataloged, even if the job abends. It can be deleted once the reorganization has been verified as successful. Extreme caution must be used to prevent the loss of valuable data. Note the use of the IF MAXCC conditional operation under IDCAMS, which will stop the reorganization if any errors occur. This is done to provide maximum protection for the data being reorganized.

**Warning: Do NOT use a temporary dataset for the sequential output file! Doing so may result in data loss should the file reload portion of the REPRO fail for any reason!**

Note that the IAM dataset is referenced by the IDS and ODS parameters. This will cause IDCAMS to dynamically allocate the IAM dataset, which is being done just in case the define results in placing the IAM dataset on a different volume.
Figure 10-75: Example of using IDCAMS to reorganize an IAM dataset (EX1081C).

//REORG  EXEC  PGM=IDCAMS,REGION=0M
//SYSPRINT DD  SYSOUT=*  
//IAMINFO DD  SYSOUT=*  
//IAMPRINT DD  SYSOUT=*  
//BACKUP DD  DSN=my.backup.iam.dataset,
//            DISP=(NEW,CATLG,CATLG),
//            UNIT=3390,VOL=SER=MY3390,
//            SPACE=(CYL,(100,50),RLSE),
// DCB=(RECFM=VB,LRECL=108,BLKSIZE=27998,BUFNO=30)
//IAMOVRID DD *
//ACCESS DD=&ALLDD,BACKUPCOMPRESSED
CREATE DD=&ALLDD,BACKUPCOMPRESSED
/*
//SYSIN DD *
LISTCAT ENT(my.iam.dataset) ALL
REPRO IDS(my.iam.dataset) OUTFILE(BACKUP)
IF MAXCC NE 0 THEN CANCEL
DELETE my.iam.dataset CLUSTER
IF MAXCC NE 0 THEN CANCEL
DEFINE CLUSTER -
   (NAME(my.iam.dataset) -
    OWNER($IAM) -
    VOL(myvol) -
    CYL(100 50) -
    RECSZ(64 100) -
    KEYS(16 0) -
    FREESPACE(5 10) -
    SHAREOPTIONS(2 3)-
    REUSE )
IF MAXCC NE 0 THEN CANCEL
REPRO INFILE(BACKUP) ODS(my.iam.dataset)
LISTCAT ENT(my.iam.dataset) ALL
/*
In this next IDCAMS example, the IAM dataset is reorganized without doing a DELETE and DEFINE. Because of that, the IAM dataset can be specified in JCL without any concerns. Note in this example that the keyword REUSE has to be specified when reloading the dataset. If this is not done, then IDCAMS does an update processing instead of load processing, and the dataset will not be reorganized.

```
//REORG EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*  
//IAMINFO DD SYSOUT=*  
//IAMPRINT DD SYSOUT=*  
//MYIAMDS DD DSN=my.iam.dataset,DISP=OLD  
//BACKUP DD DSN=my.backup.iam.dataset, 
//          DISP=(NEW,CATLG,CATLG),  
//          UNIT=3390, VOL=SER=MY3390,  
//          SPACE=(CYL,(100,50),RLSE),  
//          DCB=(RECFM=VB,LRECL=108, BLKSIZE=27998, BUFNO=30)  
//IAMOVRID DD *  
//            ACCESS DD=MYIAMDS,BACKUPCOMPRESSED  
//            CREATE DD=MYIAMDS,BACKUPCOMPRESSED  
/*  
//SYSIN DD *  
LISTCAT ENT(my.iam.dataset) ALL  
REPRO INFILE(MYIAMDS) OUTFILE(BACKUP)  
IF MAXCC NE 0 THEN CANCEL  
REPRO INFILE(BACKUP) OUTFILE(MYIAMDS) REUSE  
LISTCAT ENT(my.iam.dataset) ALL  
/*    
```

Figure 10-76: Example of Reorganization with IDCAMS, without a Delete and Define (EX1081D)
The following example demonstrates a different approach. In this example, one IAM dataset is copied over another IAM dataset with IDCAMS REPRO REUSE, resulting in a reorganized image of the original dataset. This will reduce the reorganization time by eliminating the writing to and reading from the sequential backup file. After the reorganization, renames of the datasets are done. Using this type of procedure will enable a fast reorganization to minimize the time that the data is not available. The original copy of the dataset can be used for subsequent backup or for read only processing, while normal update processing can then be resumed on the receiving dataset.

```
//REORG EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD   SYSOUT=*  
//IAMINFO DD   SYSOUT=*  
//IAMPRINT DD   SYSOUT=*  
//MYIAMDS DD   DSN=my.iam.master.dataset,DISP=OLD
//ALTIAMDS DD   DSN=my.iam.alternat.dataset,DISP=OLD
//IAMOVRID DD   * 
    ACCESS DD=MYIAMDS,BACKUPCOMPRESSED
    CREATE DD=ALTIAMDS,BACKUPCOMPRESSED
/*
//SYIN DD   *
    LISTCAT ENT(my.iam.dataset) ALL
    LISTCAT ENT(my.iam.alternat.dataset) ALL
    IF MAXCC NE 0 THEN CANCEL
    REPRO INFILE(MYIAMDS) OUTFILE(ALTIAMDS) REUSE
    IF MAXCC NE 0 THEN CANCEL
    ALTER my.iam.dataset NEWNAME(my.iam.tempname.dataset)
    IF MAXCC NE 0 THEN CANCEL
    ALTER my.iam.alternat.dataset NEWNAME(my.iam.dataset)
    IF MAXCC NE 0 THEN CANCEL
    ALTER my.iam.tempname.dataset -
    NEWNAME(my.iam.alternat.dataset)
    LISTCAT ENT(my.iam.dataset) ALL
    LISTCAT ENT(my.iam.alternat.dataset) ALL
/*
```

Figure 10-77: IDCAMS Reorganization to another IAM Dataset (EX1081E)
The example below demonstrates how to reorganize an IAM RRDS file, using a sequential file as an intermediate storage place for the data. The IAM BACKUPCOMPRESSED option MUST BE USED when performing reorganization because IDCAMS will not retain the record numbers in the sequential output dataset. With BACKUPCOMPRESSED, the record numbers will be included by IAM in the output record for the sequential file, and will likewise be present when reading the sequential file so that IAM can pick up the record number on the reload. This example does include a delete and define of the RRDS, however a REPRO REUSE could be used instead of performing the delete and define.

```plaintext
//*
//* EXAMPLE OF USING IDCAMS TO REORGANIZE AN IAM
//* RRDS FILE OR A VARIABLE RRDS FILE
//*
//REORG EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=* 
//IAMINFO DD SYSOUT=* 
//IAMPRINT DD SYSOUT=* 
//BACKUP DD DSN=MY.BACKUP.IAM.DATASET, 
// DISP=(NEW,CATLG,CATLG), 
// UNIT=3390,VOL=SER=MY3390, 
// SPACE=(CYL,(100,50),RLSE), 
// DCB=(RECFM=VB,LRECL=27994,BLKSIZE=27998,BUFNO=30) 
//IAMOVRID DD * 
//ACCESS DD=&ALLDD,BACKUPCOMPRESSED 
//CREATE DD=&ALLDD,BACKUPCOMPRESSED */ 
//SYSIN DD * 
LISTCAT ENT(MY.IAM.DATASET) ALL 
REPRO IDS(MY.IAM.DATASET) OUTFILE(BACKUP) 
IF MAXCC NE 0 THEN CANCEL 
DELETE MY.IAM.DATASET CLUSTER 
IF MAXCC NE 0 THEN CANCEL 
DEFINE CLUSTER - 
(NAME(MY.IAM.DATASET) - 
OWNER($IAM) - 
VOL(MYVOL) - 
CYL(100 50) - 
RECSZ(100 100) - 
NUMBERED - 
SHAREOPTIONS(2 3)) 
IF MAXCC NE 0 THEN CANCEL 
REPRO INFILE(BACKUP) ODS(MY.IAM.DATASET) 
LISTCAT ENT(MY.IAM.DATASET) ALL */
```

Figure 10-78: Example of RRDS Reorganization with IDCAMS (EX1081F)
In the example below, a fixed length record RRDS is reorganized directly into another fixed length record RRDS. To make sure that the target dataset has identical attributes, it is defined with the MODEL parameter specifying the input dataset. If the copy is successful, then the original dataset is deleted, and the new dataset is renamed to the original dataset name. This example is only valid for fixed length record RRDS datasets, if you need to reorganize a variable length RRDS using this method, see Example H below.

```
//**
//** EXAMPLE OF USING IDCAMS TO REORGANIZE AN IAM
//** RRDS FILE, DEFINING TARGET RRDS USING MODEL
//** THIS ONLY WORKS WITH A FIXED LENGTH RRDS FILE
//**
//REORG    EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD    SYSOUT=*
//IAMINFO DD    SYSOUT=*
//IAMPRINT DD    SYSOUT=*
//SYSIN    DD    *
DELETE MY.NEWIAM.DATASET
DELETE MY.NEWIAM.DATASET NOSCRATCH
SET MAXCC=0
LISTCAT ENT(MY.IAM.DATASET) ALL
IF MAXCC NE 0 THEN CANCEL
DEFINE CLUSTER               
  (NAME(MY.NEWIAM.DATASET)   
    OWNER($IAM)            
    MODEL(MY.IAM.DATASET))
IF MAXCC NE 0 THEN CANCEL
LISTCAT ENT(MY.NEWIAM.DATASET) ALL
REPRO IDS(MY.IAM.DATASET) ODS(MY.NEWIAM.DATASET)
IF MAXCC NE 0 THEN CANCEL
DELETE MY.IAM.DATASET CLUSTER
ALTER MY.NEWIAM.DATASET NEWNAME(MY.IAM.DATASET)
LISTCAT ENT(MY.IAM.DATASET) ALL
/*
```

Figure 10-79: Example of RRDS reorganization directly into another RRDS (EX1081G)
This example is similar to Example G above, except that this is for a variable length record RRDS. When using the MODEL sub-parameter of the DEFINE request for variable length record RRDS files, one must also specify both the RECORDSIZE parameter and the NUMBERED parameter, otherwise IDCAMS defaults to a KSDS type of dataset. For an IAM variable length record RRDS, you can leave off the RECORDSIZE parameter, however it is needed for a real VSAM RRDS.

```plaintext
/*
/* EXAMPLE OF USING IDCAMS TO REORGANIZE AN IAM
/* VARIABLE RRDS FILE, DEFINING TARGET VARIABLE RRDS USING
/* MODEL WHICH REQUIRES THAT RECORDSIZE AND NUMBERED ALSO
/* BE SPECIFIED OTHERWISE IDCAMS DEFINES
/* IT AS A KSDS
/*
//REORG EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=* 
//IAMINFO DD SYSOUT=* 
//IAMPRINT DD SYSOUT=* 
//SYSIN DD *
DELETE MY.NEWIAM.DATASET
DELETE MY.NEWIAM.DATASET NOSCRATCH
SET MAXCC=0
LISTCAT ENT(MY.IAM.DATASET) ALL 
IF MAXCC NE 0 THEN CANCEL 
DEFINE CLUSTER -
   (NAME(MY.NEWIAM.DATASET) -
    OWNER($IAM) -
    NUMBERED -
    RECORDSIZE(1100 2100) -
    MODEL(MY.IAM.DATASET))
IF MAXCC NE 0 THEN CANCEL 
LISTCAT ENT(MY.NEWIAM.DATASET) ALL 
REPRO IDS(MY.IAM.DATASET) ODS(MY.NEWIAM.DATASET)
IF MAXCC NE 0 THEN CANCEL 
DELETE MY.IAM.DATASET CLUSTER
ALTER MY.NEWIAM.DATASET NEWNAME(MY.IAM.DATASET)
LISTCAT ENT(MY.IAM.DATASET) ALL 
/*
```

Figure 10-80: Example of Variable RRDS Reorganization (EX1081H)
10.82 Deleting IAM Datasets

IAM datasets that are not members of an alternate index sphere can be deleted through any manner used to delete non-VSAM datasets, and generally any manner to delete VSAM datasets. This includes through JCL disposition (DISP) processing, various ISPF panels, including the IAM panel, the TSO DELETE command, and the IDCAMS DELETE command.

The IAM DELETE function has been enhanced as part of the alternate index support to perform dataset deletion in a manner that is compatible with VSAM. For example, when deleting a base cluster with alternate index and path associations, all of the associated datasets will be deleted. To use this function, a standard VSAM deletion be performed, rather than a non-VSAM scratch, particularly if the datasets reside on non-SMS managed volumes. Examples of VSAM deletion include the IDCAMS DELETE command, the TSO DELETE command, and the IAM ISPF delete function. Examples of functions that use the non-VSAM scratch macro include the ISPF 3.2 and ISPF 3.4 function panels. Performing an ISPF 3.2 deletion of a component of an IAM alternate index sphere will just delete that single dataset, and not the dependent associated datasets.

IDCAMS Delete

IAM will intercept any IDCAMS DELETE request to determine if the dataset is IAM or VSAM. If the dataset is an IAM dataset, then IAM will issue the system services to perform the delete request. With this feature, existing IDCAMS job steps that issue DELETE commands do not have to be changed when converting datasets to IAM. IAM makes every attempt to follow an appropriate protocol to prevent the recalling of archived or migrated datasets that are being deleted. As with any expiration date protected dataset, the PURGE keyword must be specified if the dataset is so protected. The ERASE parameter on the DELETE CLUSTER is also honored by IAM.

One difference in delete processing from VSAM is that the dataset must not be allocated to any other job or user, otherwise the delete will fail. This is because the deletion or scratch of a non-VSAM dataset internally results in z/OS issuing an exclusive ENQ.
In the example below, two IAM datasets are deleted with IDCAMS. The first one has the keyword CLUSTER specified explicitly, so IAM will handle the deletion. The second file is an IAM alternate index dataset. Even though no type is specified, IAM will still process the delete request. In addition to the alternate index, IAM will delete any path datasets that are associated with the alternate index that is being deleted. Both files are allocated in JCL with a DISP=OLD to ensure that this is the only job that is using those datasets. While those DD cards are not necessary, they may prevent deletion failures due to some other job having the dataset(s) allocated.

```
//DELETE   EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*  
//FILE1    DD   DSN=my.iam.dataset.file1,DISP=OLD
//FILE2    DD   DSN=my.iam.dataset.aix,DISP=OLD
//SYSIN    DD   *
DELETE my.iam.dataset.file1 CLUSTER PURGE
DELETE    my.iam.dataset.aix PURGE
/*
```

*Figure 10-81: Example of IDCAMS Delete of an IAM Dataset (EX1082A)*
Renaming IAM Datasets

IAM datasets can be easily renamed by using the IDCAMS ALTER command, the TSO ALTER command, or from the IAM ISPF panels. For Non-SMS managed datasets, the new dataset name remains cataloged in the same catalog as the original dataset name, even if the new high level index is an alias to a different catalog. The NONVSAM version of the ALTER command must be used for IAM datasets.

Alternate Index Concerns

IAM datasets that have alternate indexes, are alternate indexes, or are paths have special considerations due to the internally stored association information that is based on the dataset name. The association information is not automatically updated when a rename is performed. Having the correct association information is necessary to prevent failures during OPEN processing, and DELETE processing. To update the association information, it will be necessary to perform an IDCAMS DEFINE RECATALOG on one or more of the associated datasets. IAM has special processing for DEFINE RECATALOG requests to update any dataset associations. The association information can be verified by performing an IDCAMS LISTCAT ALL on the associated datasets. If you are performing renames on more than one of the associated datasets, the DEFINE RECATALOG should be executed after all of the renaming is completed.

The following rules for recataloging must be followed. First, if renaming multiple associated datasets, wait until all the renames are done before performing the recatalog. Then, the recatalog can be performed as indicated based on the type(s) of datasets that were renamed. After renaming a base cluster, all of the associated alternate indexes and paths must be recataloged. After renaming an alternate index, the alternate index and any associated paths must be recataloged. After renaming a path, only the path has to be recataloged. The recatalog must first be performed on the alternate index datasets, and then on the path datasets. Additional information on IDCAMS RECATALOG is in Section 10.84.

The IAM ISPF panels have a RENAME function, which will perform the necessary DEFINE RECATALOGS automatically, if consent is given to do so.
Renaming IAM Datasets

Example A: Rename with IDCAMS ALTER

This first example demonstrates the use of IDCAMS ALTER to rename an IAM dataset. The high level qualifier is unchanged, so the dataset will remain in the same catalog, unless you are using multilevel alias support.

```
//RENAME EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=* 
//IAMPRINT DD SYSOUT=* 
//SYSIN DD *
       LISTCAT ENT(myiam.origname.dataset) ALL
       ALTER myiam.origname.dataset
       NEWNAME(myiam.newname.dataset)
       LISTCAT ENT(myiam.newname.dataset) ALL
/*
```

Figure 10-82: Using IDCAMS ALTER to Rename an IAM Dataset (EX1083A)

Example B: Rename into Different Catalog

In this next example, an IDCAMS ALTER is again used to change the dataset name. However, in this case the dataset is a non-SMS managed dataset, and the new high level index is an alias in a different catalog than the original high level index. This problem is resolved by doing a DEFINE RECATALOG with the new dataset name, followed by deleting the new entry out of the old catalog.

```
//RENAME EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=* 
//IAMPRINT DD SYSOUT=* 
//SYSIN DD *
       LISTCAT ENT(myiam.origname.dataset) ALL
       ALTER myiam.origname.dataset
       NEWNAME(test.newname.dataset)
       DEFINE CLUSTER(NAME(test.newname.dataset) 
                      OWNER($IAM) 
                      VOL(volser) 
                      RECATALOG) 
                      CATALOG(test.catname)
       DELETE test.newname.dataset NOSCRATCH NONVSAM-
       CATALOG(myiam.catname)
       LISTCAT ENT(test.newname.dataset) ALL
/*
```

Figure 10-83: Rename into a different catalog (EX1083B)
Example C: Renaming an Alternate Index

In this example, a base cluster, the associated alternate index, and the associated path are all renamed using the IDCAMS ALTER command. Subsequent to the rename, DEFINE RECATALOG is issued on the alternate index and the path to update the association information with the new names. A volume of ANYVOL can be specified for convenience when the actual volume serial numbers are not known.

```
//RENAME EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=* 
//SYSIN    DD  *
ALTER example.myiam.cluster -
   NEWNAME(example.testiam.cluster)
ALTER example.myiam.aix -
   NEWNAME(example.testiam.aix)
ALTER example.myiam.path -
   NEWNAME(example.testiam.path)
DEFINE ALTERNATEINDEX -
   (NAME(example.testiam.aix) -
     RELATE(example.testiam.cluster) -
     VOLUME(ANYVOL) -
     RECATALOG )
DEFINE PATH   -
   (NAME(example.testiam.path) -
     PATHENTRY(example.testiam.aix) -
     RECATALOG )
LISTCAT ENT(example.testiam.cluster) ALL
LISTCAT ENT(example.testiam.aix)    ALL
LISTCAT ENT(example.testiam.path)    ALL
/*
```

Figure 10-84: Example of Rename an Alternate Index (EX1083C)
The IAM ISPF panels can also be used to rename IAM datasets, and will automatically perform the necessary recatalog. The rename function is used by selecting option R on the primary IAM panels, and indicating the name of the dataset to be renamed, along with the new name. An example of a completed rename panel is shown below, which will rename the dataset MYIAM.CLUSTER to TESTIAM.CLUSTER.

Figure 10-85: Example of IAM ISPF Rename

For IAM files that are part of an alternate index association, the following panel will appear requesting confirmation to proceed with the recatalog function. Note that in the upper right hand corner the message that the rename is completed. To proceed with the recatalog function, make sure that the Perform Recatalog is set to YES.

Figure 10-86: Example of IAM ISPF Recatalog Confirmation Panel
Figure 10-87: Recatalog confirmation screen

Upon completion of the recatalog, IAM will re-display the primary IAM panel, with a message in the upper right corner that the recatalog was ok. You can display the renamed dataset and associated components for the association information to verify that the new associations were updated properly.

Figure 10-88: Display of AIX with renamed base cluster
There are three main reasons for recataloging IAM datasets. The first reason is to rebuild the catalog entry for the dataset, if for some reason it has become uncataloged. The second reason is to update the associated dataset names for datasets that are part of an IAM alternate index sphere, if one or more of the datasets has been renamed, or copied / restored to a new dataset name. The third reason is to update certain file attributes. The recatalog process is invoked by specifying RECATALOG on the DEFINE CLUSTER, ALTERNATE INDEX, or PATH. In general only a subset of the operands is required for the recatalog process. The examples of each below will explain the required operands for each type of recatalog.

If you only need to update the catalog entry, an IDCAMS DEFINE NONVSAM request can be used. In this case, if the dataset is SMS managed, then the RECATALOG keyword must be specified, otherwise it should be left off.

In the first recatalog example, an IAM base cluster dataset is being recataloged. For this circumstance, specify the DEFINE CLUSTER command, with the following subparameters:

- Name
- OWNER($IAM), unless $IAM is in the dataset name.
- Volumes on which the dataset resides,
- And RECATALOG.

```c
//RECATLOG EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=* //IAMPRINT DD SYSOUT=* //SYSIN DD *
DEFINE CLUSTER(NAME(my.notcat.iam.dataset) -
  OWNER($IAM) -
  VOLUMES(iam001 iam002) -
  RECATALOG)
LISTCAT ENT(my.notcat.iam.dataset) ALL
/*
```

Figure 10-89: Recatalog an IAM Dataset (EX1084A)
In this example, an SMS managed IAM dataset is recataloged using a `DEFINE NONVSAM`. The following subparameters are used for this request:

- **NAME** – identifies the dataset name.
- **VOLUMES** – identifies each volume that the dataset resides on.
- **DEVT** – for each volume in the above list, a device type must be specified.
- **OWNER($IAM)** – The `OWNER` parameter is optional on the `DEFINE NONVSAM`, but is recommended so that the catalog entry will resemble the one that IAM builds.
- **RECATALOG** – This keyword must be provided if the dataset is SMS managed, otherwise it must not be specified.

```
//RECATSMS EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//IAMPRINT DD   SYSOUT=*
//SYSIN    DD   *
DEFINE NONVSAM(NAME(my.notcat.iam.dataset) -
OWNER($IAM) -
VOLUMES(iam001 iam002) -
DEVT(3390 3390) -
RECATALOG)
LISTCAT ENT(my.notcat.iam.dataset) ALL
/*
```

*Figure 10-90: Recatalog Using DEFINE NONVSAM (EX1084B)*
In this example, an IAM Alternate Index is being recataloged. The operation is quite similar to the recatalog of a base cluster, but with a few differences in the operands. The related base cluster must be specified, and the OWNER($IAM) operand is completely optional. When recataloging an alternate index, the association with the base cluster information is also updated both in the IAM alternate index dataset, and also in the IAM base cluster. The following operands are used:

- **NAME** – Indicates the name of the IAM Alternate Index.
- **RELATE** – Indicates the name of the base cluster. If the base cluster has been renamed, use the new name here. The base cluster must be cataloged prior to recataloging the alternate index.
- **VOLUMES** – Identifies the volume(s) on which the alternate index resides. If the volumes are not known and the dataset is actually cataloged, the characters ANYVOL can be specified as the volume parameter.
- **RECATALOG** – Indicates that the dataset already exists, and that the catalog entry is just being rebuilt, along with updating the base cluster association information.

```
//RECATAIX EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE ALTERNATEINDEX -
  (NAME(example.testiam.aix) -
  RELATE(example.testiam.cluster) -
  VOLUME(ANYVOL) -
  RECATALOG )
LISTCAT ENT(RAM2.TESTIAM.AIX) ALL
/*
```

*Figure 10-91: Example of Recataloging an IAM Alternate Index (EX1084C)*
Because an IAM PATH is an actual dataset, rather than just a catalog entry, the recatalog of an IAM PATH does have a special consideration, in that the PATH must be cataloged for the RECATALOG to work. This is because IDCAMS provides no mechanism to pass a volume on the DEFINE PATH command. So, if the PATH is not cataloged, it must be preceded by a DEFINE NONVSAM of the PATH, followed by the DEFINE RECATALOG. In this case, the need for the RECATALOG is to update the association information in the path itself, and the path entry dataset. The example below demonstrates how this is done. First, a LISTCAT is done on the path to see if it is cataloged. If it is not cataloged, then a DEFINE NONVSAM is issued to catalog the path. Then the DEFINE PATH RECATALOG is done.

```
//RECATPTH EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=* 
//SYSIN DD * 
LISTC ENT(RAM2.TESTIAM.PATH) 
IF LASTCC NE 0 THEN DO 
SET MAXCC=0 
DEFINE NONVSAM-( 
   NAME(RAM2.TESTIAM.PATH) - 
   VOLUMES(JUNK01) - 
   DEVT(3390) - 
   OWNER($IAM) ) 
END 
ELSE 
DEFINE PATH-( 
   NAME(RAM2.TESTIAM.PATH) - 
   PATHENTRY(RAM2.TESTIAM.AIX) - 
   RECATALOG ) 
LISTCAT ENT(RAM2.TESTIAM.PATH) ALL
/*
```

Figure 10-92: Example of Recatalog an IAM Path (EX1084D)
IAM itself does not do any processing for the ALTER command however various file attributes can now be changed through the DEFINE RECATALOG process. When executed with the use of the IAM CREATE override statement certain attributes can be revised including buffer space, minimum and maximum buffer specifications for a dataset, journaling options, the share options, and the RLSID. In the example below, the RLSID is nullified, and the share options are changed to 2. By specification of a volume of ANYVOL, the actual volume(s) on which the dataset resides does not need to be known.

```plaintext
/*
/*
/*  EXAMPLE OF REMOVING RLSID AND CHANGING SHAREOPTION
/*
/*
//EX1084E EXEC PGM=IDCAMS
//IAMPRINT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//IAMOVRID DD *
CREATE DD=&ALLDD,RLSID=,SHARE=2
/*
//SYSIN DD *
DEFINE CLUSTER -
  (NAME(IAMV.EX1084E.CLUSTER) -
   VOL(ANYVOL) -
   OWNER($IAM ) -
   RECATALOG)
LISTC ENTRIES(IAMV.EX1084E.CLUSTER) ALL
/*
```

Figure 10-93: Altering an IAM File's Attributes (EX1084E)
10.85  Backing up and Restoring IAM Datasets

Backup Overview

Frequently, IAM datasets are incorporated into normal DASD management procedures for backup, restore, and archival (or migration) purposes. Software products that provide such services can easily handle IAM datasets within the normal functionality that they provide. The major considerations are first, that IAM datasets are treated as non-VSAM datasets by these products, and second that the default DSORG of PS for IAM files is highly recommended to provide the most flexibility when managing IAM datasets with such products. The functionality and limitations that these products have with non-VSAM datasets will apply for IAM datasets.

Many application job streams incorporate their own backup of their related datasets in scheduled time frames that are more relevant to the application, and provide for improved recoverability of application data. As an aid to those responsible for maintaining recoverability of application data contained in IAM datasets, this section will present some of the considerations with the various methods of backing up and restoring at the dataset level, along with some examples. The examples are intended to demonstrate basic functionality of the various backup and restore methods that are applicable to IAM datasets.

Alternate Index Considerations

For IAM datasets in an associated alternate index sphere, as with VSAM, the most critical piece is the base cluster dataset. After restoring a base cluster, the alternate indexes and paths can always be recreated if backups are not readily available. The major concern with IAM alternate indexes and their associated datasets is that DASD management utility programs will treat each dataset as a separate entity. With VSAM, some of these utilities have developed processing to recognize the associations, and handle the base cluster and all associated alternate indexes and paths as one entity. Using dataset naming conventions within a set of alternate index associated datasets where the only name difference is in the last index level of the name will help ease the dataset management difficulties. By doing so, you can treat any set of associated datasets by a dataset group name in the control cards for the utility. Information on how to do this will be provided in some of the subsequent examples.

Backup with IDCAMS

The most common method for backing up IAM base cluster and alternate index datasets by application job streams is with IDCAMS REPRO. This method copies the data at the record level into a sequential output dataset. The main advantages of using such a backup are that application programs can directly read the data from the backup sequential dataset, a restore will reorganize the dataset, and when moved to a different device type IAM will adjust the block size automatically to obtain the best possible device utilization. Performance of such a backup can be very fast through effective use of buffering. If the sequential copy of the data is only used for backup purposes, then the IAM Backup Compressed feature can be used if the dataset was compressed with IAM software compression. This feature prevents the decompressing of the data during backup, and the compressing of data when it is used to reload the dataset. Using the IAM Backup Compressed feature saves CPU time, as well as I/O time transferring data to and from the backup media. There is unlikely to be any savings in terms of the number of tapes required to backup the IAM dataset when the tape devices that also offer compression, because the data is already compressed. The main disadvantage of IDCAMS REPRO is that the attributes used to define the dataset are not retained with the output dataset. While IDCAMS does offer a function for VSAM datasets to save the dataset attributes, called EXPORT, and the corresponding restoration of dataset and attributes with IMPORT, IAM does not provide support for the use of those functions.
An alternative to using IDCAMS is to use a software product that provides DASD dataset management functions, including backing up and restoring datasets. Examples of such products include FDR, ABR, DFSMSdss, and DFSMSHsm. These products offer the fastest possible data movement, along with various dataset management capabilities. A major advantage of using one of these products is that the file attribute information is saved with the backup, along with an exact image of the dataset. The disadvantages are that application programs can not directly access the data because of the format it is stored in, the software is not as easy to use as an IDCAMS REPRO, when restored the dataset is not reorganized, and restoration to an unlike device can cause inefficient usage of DASD space. However, when fast backup times are required, and when the backup copy of the data is not being used by the application, then these DASD utilities will meet that requirement.

The use of IDCAMS REPRO for backing up and restoring IAM datasets is straightforward. Jobs that have been set up to backup VSAM datasets with this method should not require any changes. One thing to be careful of is that IDCAMS defaults sequential output files to RECFM=U, which forces each output record to become a physical block on the storage media. This can waste media and backup time. By assigning the output dataset a RECFM=VB, and a block size of 27998 for 3390 DASD, or 32760 for tape, plus providing a large number of buffers, will speed up the REPRO process substantially. When an installation uses the IAM buffering defaults as shipped, then there should be no need for IAM overrides on an IDCAMS repro, unless you desire to use BACKUPCOMPRESSED.
Example A: IDCAMS Backup

In this example, a large multi-volume IAM dataset is backed up by an IDCAMS REPRO. The output media is tape. Note that the DCB information that is provided including BUFNO. The LRECL for the output tape volume is the maximum record size defined for the IAM dataset plus four for the RDW added for variable format records and an additional four for IAM compression data. So, assuming the IAM file has a maximum record length of 1016, a value of at least 1024 must be used for the LRECL on the output dataset. The BACKUPCOMPRESSED override is specified because the sequential output file will not be used for any purpose other than to reload the dataset should a restore be needed. A volume count of 20 is provided for the output tape dataset to provide enough tape volumes to hold the data. A LISTCAT ALL is included so that information on what is needed to define the dataset, if necessary, is readily available.

```
//BACKUP   EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*  
//IAMINFO DD   SYSOUT=*  
//IAMPRINT DD   SYSOUT=*  
//BACKUP   DD   DSN=my.iam.dataset.backup,DISP=(,CATLG),    
//   DCB=(RECFM=VB,LRECL=1024,BLKSIZE=32760,BUFNO=30),  
//   UNIT=3490,VOL=(,,20)   
//IAMFILE DD   DSN=my.iam.dataset,DISP=OLD   
//IAMOVRID DD   *       
//ACCESS DD=&ALLDD,BACKUPCOMPRESSED   
/*   
//SYSIN   DD   *  
LISTCAT ENT(my.iam.dataset) ALL   
REPRO INFILE(IAMFILE) OUTFILE(BACKUP)   
/*
```

Figure 10-94: Example of IDCAMS Backup (EX1085A)
Example B: IDCAMS Restore

The following example demonstrates how to restore the dataset that was backed up in the prior example. The IAM dataset is not being deleted and redefined, so the REUSE parameter is required on the REPRO statement. A BUFNO is provided on the tape input to speed up reading the data. A CREATE override indicating BACKUPCOMPRESSED is required because the backup tape contains IAM compressed data. A LISTCAT ALL is done after the REPRO to verify the new file structure.

```plaintext
//RESTORE EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=* 
//IAMPRNT DD   SYSOUT=* 
//IAMINFO DD   SYSOUT=* 
//BACKUP   DD   DSN=my.iam.dataset.backup,DISP=OLD, DCB=BUFNO=30,UNIT=3490 
//IAMFILE DD   DSN=my.iam.dataset,DISP=OLD 
//IAMOVRID DD   * 
   CREATE DD=&ALLDD,BACKUPCOMPRESSED */ 
//SYSIN    DD   * 
   REPRO INFILE(BACKUP) OUTFILE(IAMFILE) REUSE 
   LISTCAT ENT(my.iam.dataset) ALL */ 
```

Figure 10-95: Example of an IDCAMS Restore with BACKUPCOMPRESSED (EX1085B)

Example C: Decompress the Backup Dataset

If ever needed, the IAMRECVR utility program can be used to create a sequential uncompressed file from the sequential file with records in an IAM data compressed format, as created by using the BACKUPCOMPRESSED override. The new file will contain uncompressed records from the original dataset, ready to be used by application or other processing needs. To perform this function, you must know the key length and key offset (RKP) for the file being processed. These values can be obtained from a LISTCAT or an IAMINFO report from the original dataset. In the example below, a new sequential backup with uncompressed records is created using the sequential dataset created in Example EX1085A. The DISKIN DD specifies the original backup dataset, and the TAPEOUT DD specifies the new uncompressed dataset.

```plaintext
//DECOMPRS EXEC PGM=IAMRECVR,REGION=0M 
//SYSPRINT DD   SYSOUT=* 
//DISKIN   DD   DSN=my.iam.dataset.backup,DISP=OLD 
//TAPEOUT  DD   DSN=my.iam.dataset.uncomp,UNIT=TAPE, DISP=(,CATLG), DCB=(RECFM=VB,LRECL=1020,BLKSIZE=32760) 
//SYSIN    DD   * 
   DECOMPRESS    KEYLEN=16,RKP=0 */ 
```

Figure 10-96: Example of decompressing a BACKUPCOMPRESSED Dataset (EX1085C).
For the DASD backup and restore utilities, the examples will be split into single volume datasets, and multivolume datasets. Single volume IAM datasets are very easily dumped and restored with either FDR or DFSMSdss. While normally not necessary, it is recommended that the original dataset be deleted prior to doing the restore if you are replacing the original dataset. When the delete is done, the restore will allocate the dataset attempting to obtain all the required space in one extent, which may be beneficial because the file may be able to acquire more DASD space.

The single volume examples shown below are all set to handle either a single IAM dataset, or all associated IAM datasets in an alternate index sphere, if they are all on the same volume. If the various associated datasets are on different volumes, then use the multivolume procedures. For these examples, each associated dataset in the alternate index sphere has an identical high level qualifier(s), with only the last level being different. Assume the following names:

- Base Cluster DSN=my.iam.dataset.cluster
- Alternate Index DSN=my.iam.dataset.aix
- Path DSN=my.iam.dataset.path

If you are using different naming conventions, refer to the manual for the product you use for further information on building a dataset name mask.

In this first example, FDR is being used to backup an IAM dataset to tape. The JCL requirements are for a DISKx DD statement that specifies the volume of the dataset being backed up. Then a corresponding TAPEx DD statement is required for the tape to contain the backup data.

```jcl
//BACKUP   EXEC PGM=FDRDSF,REGION=0M
//SYSPRINT DD   SYSOUT=*  
//DISK1    DD   DSN=my.iam.dataset.cluster,DISP=OLD
//TAPE1    DD   DSN=my.iam.dataset.backup,  
//          DISP=(,CATLG),UNIT=TAPE
//SYSIN    DD   *  
DUMP TYPE=DSF  
S DSN=my.iam.dataset.**  
/*
```

*Figure 10-97: Example of FDR/DSF Backup of Single Volume IAM Dataset (EX1085D)*
Example E: FDR/DSF Restore

The restore of the dataset is just as easy. This example works whether or not the original dataset had been deleted. The DISK1 DD specifies the receiving volume, which does not have to be the same as the dataset originally resided on. The TAPE1 DD specifies the backup tape created in the prior backup example. The backup tape could also be from a full volume backup as well.

```
//RESTORE EXEC PGM=FDRDSF,REGION=0M
//SYSPRINT DD   SYSOUT=*  
//DISK1   DD   UNIT=3390,VOL=SER=iamvol,DISP=OLD
//TAPE1   DD   DSN=my.iam.dataset.backup,DISP=OLD
//SYSIN   DD   *        
RESTORE TYPE=DSF
   S DSN=my.iam.dataset.**
/*
```

Figure 10-98: Example of an FDR/DSF restore of a Single Volume IAM Dataset (EX1085E)

Example F: DFSMSdss Backup

In this next example, DFSMSdss is being used to backup an IAM dataset. The output tape is specified by the OUTDD parameter, and use of an input DD is optional, and not shown in this example.

```
//BACKUP   EXEC PGM=ADRDSSU,REGION=0M
//SYSPRINT DD   SYSOUT=*  
//TAPE1    DD   DSN=my.iam.dataset.backup,  
//            DISP=(,CATLG),UNIT=TAPE
//SYSIN    DD   *        
DUMP OUTDD(TAPE1) DS(INCL(my.iam.dataset.**))  
/*
```

Figure 10-99: Example of DFSMSdss Dump of a Single Volume IAM Dataset (EX1085F)

Example G: DFSMSdss Restore

This corresponding example demonstrates how to restore the IAM dataset. The output volume could be the same as the original, or different.

```
//RESTORE EXEC PGM=ADRDSSU,REGION=0M
//SYSPRINT DD   SYSOUT=*  
//DISK1   DD   UNIT=3390,VOL=iamvol,DISP=OLD
//TAPE1   DD   DSN=my.iam.dataset.backup,DISP=OLD
//SYSIN   DD   *        
RESTORE DS(INCL(my.iam.dataset.**)) -  
   INDD(TAPE1) OUTDD(DISK1) REPLACE CATALOG
/*
```

Figure 10-100: Example of DFSMSdss Restore of a Single Volume IAM Dataset (EX1085G)
Multivolume backup and restore operations require a little more effort. The best overall process for the combined backup and restore effort seems to be using the application backup capabilities of ABR. While this does require an additional job step on both the backup and restore jobs, the dataset is restored across the same number of volumes as it had originally, along with the same amount of space on each volume. If the ABR product is not available, then the same results can still be achieved with FDR/DSF with a little more effort on the JCL. DFSMSdss can also be used for multivolume backup and restore, although it may be difficult to restore the dataset spread across multiple volumes as it was originally.

The multivolume backup and restore process can also be used when backing up an IAM alternate index sphere, where the datasets reside on different volumes, or one or more of the datasets resides on multiple volumes. These set of examples will continue to assume use of the dataset naming conventions from the single volume examples.

Example H: ABR Backup

In this example, a multivolume IAM dataset will be backed up using ABR application backup and recovery capability. While the example given here is just a single dataset, multiple datasets for the entire application can be backed up and restored as necessary with this technique. Basically, this process involves creating a temporary archive control file, which is used to store information about the dataset(s) backed up, and is subsequently also backed up to the same tape as the dataset(s). If this type of backup job is being set up to run on a regular basis, then GDG dataset names can be used for both the Archive control file, and the backup of the Archive control file. Note that ABR requires that the word ARCHIVE be included within the dataset name for the Archive control file. The name of the backup of the Archive control file will be similar, only with ARCBKUP in place of ARCHIVE. The Archive control file can be deleted after this job runs. Full details on using this capability of ABR can be found in section 52 of the FDR manual.

```
//INITARCH EXEC PGM=FDRARCH
//SYSPRINT DD SYSOUT=*  
//ARCHIVE DD DSN=my.ARCHIVE.dataset,UNIT=SYSDA,
//         DISP=(,CATLG),SPACE=(TRK,(5,5),RLSE)
//SYSIN DD *
//    FORMAT RECS=nnnn,USERINDEX=YES  set to number of datasets
/*
//BACKUP EXEC PGM=FDRABR,REGION=0M
//SYSPRINT DD SYSOUT=*  
//SYSPRINAA DD SYSOUT=*  
//ARCHIVE DD DSN=my.ARCHIVE.dataset,DISP=OLD
//TAPEA DD DSN=my.backup.dataset,
//UNIT=TAPE,DISP=(,KEEP)
//SYSIN DD *
// DUMP TYPE=APPL,ONLVOL,RTC=YES,
// ARCBKUP=DSF,DSNENQ=HAVE
// SELECT CATDSN=my.iam.dataset.**
/*
```

Figure 10-101: Example of FDR/ABR Backup of Multivolume IAM Dataset (EX1085H)
Example I: ABR Restore

The restore process is a two step process as well. The first step restores the copy of the archive control file that was backed up, then the next step restores the IAM dataset. In this example the dataset is being restored to the original volumes, however there are various options to change the volumes that it is being restored to, as well as to give the dataset a new name. When restoring as the original dataset, it is highly recommended that the dataset be deleted first, so that ABR can allocate it as it existed previously.

```plaintext
//DELETE   EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*  
//SYSIN DD   *
   DELETE my.iam.dataset.cluster
   DELETE my.iam.dataset.cluster NOSCRATCH
   SET MAXCC=0
/*
//RESTARC  EXEC PGM=FDRABR,REGION=0M
//SYSPRINT DD   SYSOUT=*  
//SYSPRINA DD   SYSOUT=*  
//TAPEA    DD   DSN=my.ARCBKUP.dataset,  
//           VOL=(,RETAIN),DISP=OLD
//SYSIN DD   *
   RESTORE  TYPE=ABR,RECAT
   SELECT   DSG=my.ARCHIVE,TAPEDD=A
/*
//RESTDATA EXEC PGM=FDRABR,REGION=0M
//SYSPRINT DD   SYSOUT=*  
//ARCHIVE  DD   DSN=my.ARCHIVE.dataset,DISP=OLD
//TAPE1    DD   DSN=my.FDR,DISP=OLD,VOL=REF=*.*.RESTARC.TAPEA
//SYSIN DD   *
   RESTORE TYPE=APPL,RECAT
   SELECT DSN=my.iam.dataset.**
/*
```

Figure 10-102: Example of FDR/ABR Restore of Multivolume IAM Dataset (EX1085I)
FDR/DSF can also be used to backup and restore a multivolume IAM dataset, although the use of ABR Application backup is strongly recommended. A three volume IAM dataset is being backed up in this example. Each volume is effectively a separate backup, but can and should be done in one job step. For each DASD volume, there will be a DISKx and corresponding TAPEx DD card. As long as you make sure that you get all the pieces on both the backup and restore, this process will work fine.

```
//BACKUP   EXEC PGM=FDRDSF,REGION=0M
//SYSPRINT DD   SYSOUT=*  
//DISK1    DD   UNIT=3390,DISP=OLD,VOL=SER=iam001
//DISK2    DD   UNIT=3390,DISP=OLD,VOL=SER=iam002
//DISK3    DD   UNIT=3390,DISP=OLD,VOL=SER=iam003
//TAPE1    DD   DSN=my.iam.backup.vol1, 
//          UNIT=TAPE,DISP=(,CATLG)
//TAPE2    DD   DSN=my.iam.backup.vol2, 
//          UNIT=TAPE,DISP=(,CATLG)
//TAPE3    DD   DSN=my.iam.backup.vol3, 
//          UNIT=TAPE,DISP=(,CATLG)
//SYSIN    DD   *  
DUMP     TYPE=DSF
SELECT   DSN=my.iam.dataset.**
/*
```

*Figure 10-103: Example of using FDR/DSF to Backup a Multivolume IAM Dataset (EX1085J)*
To restore the dataset, the procedure is almost identical. In this example, the dataset is being restored to the same three volumes, however different volumes could be used, and the dataset could be given a new name. If the dataset is being restored to replace the existing version, it is required that the dataset be deleted prior to performing the restore. To insure that the dataset does not exist, the restore is preceded by an IDCAMS step to delete the dataset.

```
//DELETE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
   DELETE my.iam.dataset.cluster
   DELETE my.iam.dataset.cluster NOSCRATCH
   SET MAXCC=0
/*
//RESTORE EXEC PGM=FDRDSF,REGION=0M
//SYSPRINT DD SYSOUT=* 
//DISK1 DD UNIT=3390,DISP=OLD,VOL=SER=iam001
//DISK2 DD UNIT=3390,DISP=OLD,VOL=SER=iam002
//DISK3 DD UNIT=3390,DISP=OLD,VOL=SER=iam003
//TAPE1 DD DSN=my.iam.backup.vol1,DISP=OLD
//TAPE2 DD DSN=my.iam.backup.vol2,DISP=OLD
//TAPE3 DD DSN=my.iam.backup.vol3,DISP=OLD
//SYSIN DD *
   RESTORE TYPE=DSF
   SELECT DSN=my.iam.dataset.**
/*
```

Figure 10-104: Example of Using FDR/DSF to Restore a Multivolume IAM Dataset (EX1085K)

In this last pair of examples, DFSMShsm is being used to backup and subsequently restore a multivolume IAM dataset. As can be seen, the backup portion of this example is essentially identical to the backup of a single volume IAM dataset.

```
//BACKUP EXEC PGM=ADRDSSU,REGION=0M
//SYSPRINT DD SYSOUT=* 
//TAPE1 DD DSN=my.iam.backup,UNIT=TAPE,DISP=(,CATLG)
//SYSIN DD *
   DUMP OUTDD(TAPE1) DS(INCL(my.iam.dataset.**))
/*
```

Figure 10-105: Example of DFSMSdss Backup of a Multivolume IAM Dataset (EX1085L)
Now, in this example DFSMSdss will be used to restore the dataset backed up in the prior example. Unless you can be certain that the IAM dataset was not deleted and defined or reorganized since the backup was done, the dataset must be deleted prior to attempting the restore. This is to prevent a possible restore failure. To force the dataset to be split across the three volumes, temporary datasets are allocated on all of the DASD volumes so that DFSMSdss will not find enough space to restore the whole dataset on any particular volume. The manual part of this process involves checking each volume for how much space it has available, and determining how much space is needed by the dataset being restored, and how much space should be used on each volume for the dataset being restored. Then, set up the JCL to allocate a temporary dataset on each volume with the remaining free space.

```
//DELETE   EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*  
//SYSIN    DD   *  
   DELETE my.iam.dataset.cluster  
   DELETE my.iam.dataset.cluster NOSCRATCH 
   SET MAXCC=0  
/*  
//RESTORE  EXEC PGM=ADRDSSU,REGION=0M
//SYSPRINT DD   SYSOUT=*  
//TEMP1    DD   UNIT=3390,VOL=SER=iam001,SPACE=(CYL,2000)  
//TEMP2    DD   UNIT=3390,VOL=SER=iam002,SPACE=(CYL,1500)  
//TEMP3    DD   UNIT=3390,VOL=SER=iam003,SPACE=(CYL,400)  
//TAPE1    DD   DSN=my.iam.backup,DISP=OLD  
//SYSIN    DD   *  
   RESTORE  DATASET(INCLUDE(my.iam.dataset.**)) -  
       INDD(TAPE1) ODY((iam001),(iam002),(iam003)) -  
       REPLACE CATALOG  
/*
```

Figure 10-106: Example of DFSMSdss Restore of a Multivolume IAM Dataset (EX1085M)
10.86 Moving IAM Datasets

There are several different software utilities that can be used to move IAM datasets. When IAM datasets are being moved along with many other datasets, for example as part of a reconfiguration, DASD backup and restore software, such as FDR from Innovation or DFSMSdss from IBM are frequently used. Other related options are to move the dataset(s) with FDRCOPY, or with FDRREORG.

Moving with FDRREORG

The best way from an internal dataset structure and DASD space view point, is to move IAM datasets by performing dataset reorganization. FDRREORG provides an excellent tool for moving IAM dataset(s) to a different volume(s). FDRREORG not only knows the basic file attributes, it also knows what IAM Overrides were used to create the original dataset, and will pass those override values when defining the new dataset. The other major alternative is to use IDCAMS REPRO, although that does involve more manual effort than FDRREORG.

Example A: Using FDRREORG to Move IAM Datasets

Shown below is an example of moving all of the IAM datasets from one volume to another. The NODEFAULTS keyword means that the selection criteria are based entirely on the control card input. The NOUPDATES=YES causes the datasets to be reorganized, even if there were no updates, inserts, or deletes. The IAMDEFINE=YES forces the delete and define of the IAM datasets. On the Select card all IAM datasets are selected from one volume, and moved to another volume. The NEWALL keyword is required so that the new volume will be applied to all of the selected datasets.

```
//IAMDSMOV EXEC PGM=FDRREORG,REGION=0M
//SYSPRINT DD   SYSOUT=*  
//REORGPRT DD   SYSOUT=* 
//REORGRPT DD   SYSOUT=*  
//IAMINFO  DD   SYSOUT=*  
//SYSIN    DD   *  
REORG    NODEFAULTS,NOPUPDATES=YES,IAMDEFINE=YES
SELECT   ALLDSN,DSTYPE=IAM,VOL=oldvol, 
         NEWVOLSDATA=newvol,NEWALL  
/*
```

Figure 10-107: Example of Using FDRREORG to Move IAM Datasets (EX1086A)
Example B: Using FDREPORT to Generate IDCAMS Control Cards

Another alternative automation technique for moving of IAM datasets by reorganization is to use FDREPORT. As demonstrated below FDREPORT can be used to create a dataset containing control cards for IDCAMS for the following functions:

1. Define an IAM dataset on the new volume modeled from the original dataset.
2. Copy the original dataset into the new dataset.
3. Delete the original dataset.
4. Rename the new dataset to the original dataset name.
5. Do a LISTCAT on the moved dataset.

If an error occurs then the IDCAMS execution is stopped. The model control cards for each dataset are specified by the data in the MASK DD input stream. By control card specifications to FDREPORT only IAM datasets are selected that are single volume datasets cataloged to the volume that has been selected. One could get more sophisticated with the IF logic control cards created for IDCAMS so that processing will resume with the next dataset rather than stopping completely on the first error.

```
//SELECTDS EXEC PGM=FDREPORT,REGION=0M
//SYSPRINT DD SYSOUT=*  
//SYSPUNCH DD DSN=&&CAMS,UNIT=SYSDA,SPACE=(CYL,(2,1)), DISP=(,PASS),  
//                  DCB=(RECFM=FB,LRECL=80,BLKSIZE=23440)
//SYSIN DD *  
XSELECT DSORG=IAM,VOL=oldvol,CATALOG=YES,CATVOLCT=1 
PUNCH FDRLIB=MASK  
PRINT ENABLE=IAM,RPTYPE=SELPCH,SORT=NO
/*
//MASK DD *  
DEFINE CLUSTER(NAME(<DSN>).NEW) -   
    MODEL(<DSN>) -  
    OWNER($IAM) -  
    VOL(newvol) )   
IF MAXCC NE 0 THEN CANCEL  
REPRO IDS(<DSN>) ODS(<DSN>.NEW) 
IF MAXCC NE 0 THEN CANCEL  
DELETE <DSN> 
ALTER <DSN>.NEW NEWNAME(<DSN>) 
IF MAXCC NE 0 THEN CANCEL  
LISTCAT ENT(<DSN>) ALL 
)SUFFIX 
IF MAXCC NE 0 THEN CANCEL  
/*
//MOVEIAM EXEC PGM=IDCAMS,COND=(0,NE) 
//SYSPRINT DD SYSOUT=*  
//IAMPRINT DD SYSOUT=* 
//IAMINFO DD SYSOUT=* 
//SYSSIN DD DSN=&&CAMS,DISP=OLD
```

Figure 10-108: Example of using FDREPORT to Generate Control Cards for IDCAMS (EX1086B)
Moving IAM Datasets

FDRCOPY can also be used to move IAM datasets. The advantage of using FDRCOPY is that all or a selected subset of datasets can be copied, regardless of type, in one job step. The disadvantage, as discussed above, is that IAM datasets will not be reorganized or reblocked. However, it is quite easy to use. In the following example all non-VSAM datasets are being moved, which will include any IAM datasets.

Example C: Moving Datasets with FDRCOPY

FDRCOPY can also be used to move IAM datasets. The advantage of using FDRCOPY is that all or a selected subset of datasets can be copied, regardless of type, in one job step. The disadvantage, as discussed above, is that IAM datasets will not be reorganized or reblocked. However, it is quite easy to use. In the following example all non-VSAM datasets are being moved, which will include any IAM datasets.

```plaintext
//MOVENVSM EXEC PGM=FDRCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
MOVE TYPE=DSF  
SELECT ALLDSN,VOL=oldvol,NVOL=newvol
/*
```

Figure 10-109: Example of Moving Datasets with FDRCOPY (EX1086C)

Example D: Moving an IAM Dataset with FDRCOPY

You can also use FDRCOPY to move a single dataset or group of IAM datasets. In the example below, a single IAM dataset is moved with FDRCOPY to a new volume, as identified by the NVOL parameter.

```plaintext
//MOVEDS EXEC PGM=FDRCOPY,REGION=0M  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
MOVE TYPE=DSF  
SELECT CATDSN=my.iam.dataset,NVOL=newvol
/*
```

Figure 10-110: Example of Moving Datasets with FDRCOPY (EX1086D)

Example E: Moving Datasets with DFSMSdss

A similar type of dataset move function can also be accomplished with DFSMSdss. The move operation is effected by specifying the DELETE and RECATALOG(*) keywords on the control card. Non-VSAM datasets with organizations of sequential, partitioned, or direct are being moved. The AUTORELBLKA keyword is specified to insure that IAM datasets with DSORG=DA are copied properly.

```plaintext
//KSD972C EXEC PGM=ADRDSSU,REGION=0M,COND=EVEN  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
COPY DATASET (INCLUDE(**) -  
   BY((DSORG EQ (SAM,PDS,BDAM)) (CATLG EQ YES)) ) -  
   LIDY((oldvol)) ODY((newvol)) -  
DELETE RECATALOG(*) AUTORELBLKA
/*
```

Figure 10-111: Example of Moving Datasets with DFSMSdss (EX1086E)
Recovering IAM Datasets

As with any dataset containing critical data, it is important to have established dataset recovery procedures in place. This includes having proper dataset backups and a way to restore any potential data that may be lost due to a hardware or software failure. Most software packages that provide logging and recovery for VSAM files can also be used with IAM datasets. To assist in dataset recovery IAM provides the utility program IAMRECVR that may be of assistance in the recovery process. IAMRECVR is not intended to take the place of established dataset recovery procedures, but may aid in reducing the time required to recover a dataset, or prevent the use of such procedures in some circumstances. IAMRECVR can only retrieve the data that physically exists on the DASD device in a readable form. Data that is inaccessible due to media or other failures are not recoverable with this utility. Likewise, data that has been overwritten or never written out to the storage media cannot be recovered by IAMRECVR.

IAM Alternate index datasets can be recovered with either the IAMRECVR utility, or by rebuilding the alternate index using the IDCAMS BLDINDEX function. Using IDCAMS may be preferable, particularly when there is potential data loss, such as may occur with a hardware data check, or other media failure.

IAMRECVR

Should you suspect any type of problem with an IAM file, either physical (I/O errors), or software failures, IAMRECVR can be used to aid in diagnosing and recovering IAM datasets from such types of errors. IAMRECVR offers a set of services that includes the following commands:

1. DIAGNOSE - Read the entire dataset to validate general dataset integrity.
2. PRINT - Print selected portions of the IAM dataset to provide diagnostic information.
3. RECOVER - Reads all of the data in the IAM dataset, and copies the readable data to a sequential output dataset that can be used to reload the file. Records with duplicate keys are optionally written to a separate sequential dataset.
4. APPLY - Copies data from the duplicate log file into the newly loaded recovered dataset.

IAMRECVR has knowledge about underlying structure of an IAM dataset, and reads IAM files without using the IAM access method. IAMRECVR utilizes high performance I/O, reading up to an entire cylinder per physical I/O. With the information about each particular dataset, which is usually retrieved from the dataset itself, IAMRECVR pulls out data records from the file. While the entire dataset will be read, data blocks containing the index structure are not processed.

This section will explain how to validate dataset integrity, how to collect any diagnostic information that may be requested, and how to recover the dataset from various error situations through the use of IAMRECVR and various other utilities. Complete information on all of the functions and operands available with IAMRECVR are provided in the System Analysis Utilities Section 45 of the manual. The majority of dataset integrity problems are the result of improper dataset sharing. In those circumstances, there may quite likely be some data lost because it may have been overwritten, and cannot be retrieved because it is not there. Complete data recovery in those situations will require the use of additional recovery procedures beyond those provided by IAM.
There are two general categories of failures. The first category consists of errors encountered processing the dataset. For example, you may have a program that is receiving an IAMW12 message indicating a data decompression error, or perhaps an IAMW37 I/O error message, or an IAMW17 error message indicating concurrent updates. Or perhaps you are receiving unexpected errors or results from the application, which can include receiving unexpected error codes from IAM while processing the dataset. These types of errors may indicate a problem with the dataset itself, however they more frequently arise due to software failures such as inadvertent storage corruption. Such errors frequently do not corrupt the file. For these situations, one would normally start by running an IAMRECVR DIAGNOSE operation to validate general dataset integrity, which then may be followed with a recovery or a reorganization of the dataset. If you suspect that IAM is not returning all of the records that are expected, it is critical to run an IAMRECVR RECOVER on the dataset before reorganizing the file or otherwise copying the data. Depending on the nature of the problem, IAMRECVR may be able to retrieve records that cannot be retrieved by normal IAM processing that utilizes the index structure of the dataset.

The second category of failure is where IAM is unable to open the dataset. Generally, an IAM error message will be displayed indicating the cause of the failure to open. If the problem is other than environmental, such as insufficient storage, then a dataset recovery will be necessary. Generally, this type of error is accompanied by an IAMW79 or an IAMW01 with an IAMW37 error message. These messages contain information about the point of failure during the open, so it is critical to save this message for diagnostic purposes. For this category of errors it is best to start out with obtaining initial diagnostic information, and then proceed directly to use the IAMRECVR RECOVER command to copy the data records out of the file.

IAMRECVR will provide a report on any errors it may find, plus it will indicate how many records that it actually read. This record count can be used to compare with the count from other sources to determine if all the records have been retrieved. There may be some circumstances where an IAM dataset cannot be opened for which IAMRECVR successfully performs the recovery, however finds no detectable errors. This is because IAMRECVR is not using the index structure of the dataset to read the data, so it may not be detecting the error condition.
When errors are encountered when processing an IAM dataset, generally the first step is to validate the general dataset integrity with an IAMRECVR DIAGNOSE function. This is most useful for circumstances where you suspect that there might be a problem with an IAM dataset. For example, if a job has received an IAMW12 data decompress error, or an IAMW17 indicating concurrent updating, or an I/O error message processing the IAM dataset. Or, perhaps there has been a system failure and you just want to verify the integrity of the IAM dataset. This is done with the DIAGNOSE command.

DIAGNOSE will read the entire IAM dataset, validating basic file integrity. It will verify the following:

1. All data blocks can be physically read.
2. Records are in ascending key sequence within each data block.
3. All data blocks have valid structure.
4. All compressed records can be decompressed.
5. Records are in ascending key sequence within the prime area of the dataset.
6. Verify that there are no duplicate records within a data block.

DIAGNOSE will provide information on any errors it detects, along with a count of the number of records it was able to read from the dataset. The output will also include a report with basic information about the dataset that is quite similar to the IAMPRINT LISTCAT report. This information will help you identify if there is actually a problem within the dataset itself, and if so how much of the data will be recoverable.

Example A: IAMRECVR DIAGNOSE

The example below demonstrates how to run an IAMRECVR DIAGNOSE. The requirements are providing a control card input on SYSIN that specifies the DIAGNOSE command, a DD statement with the name of DISKIN that allocates the IAM dataset to be diagnosed, and a SYSPRINT DD card for the printed output.

```plaintext
//DIAGNOSE EXEC PGM=IAMRECVR,REGION=0M
//SYSPRINT DD SYSOUT=*  
//DISKIN DD DSN=my.iam.dataset,DISP=SHR
//SYSIN DD *
DIAGNOSE
/*
```

The results of the DIAGNOSE process will provide information on the status of the dataset, and how many of the data records can be recovered. If problems were found, or difficulties are continuing with the dataset, then a recovery must be performed. Based on the information provided by the DIAGNOSE run, you can then determine what the best recovery method will be for this particular dataset. If it looks like IAMRECVR can retrieve all of the data records, or there are no other alternative recovery procedures available, then the IAMRECVR RECOVER command can be used to create a sequential output file containing the data.
IAMRECVR can also DIAGNOSE multiple datasets in a single execution. This is achieved by specifying different DD names on separate DIAGNOSE commands using the FROMDDNAME keyword on the DIAGNOSE command, as illustrated by the following example.

```
//MULTDIAG EXEC PGM=IAMRECVR,REGION=0M
//SYSPRINT DD SYSOUT=*  
//FILE1 DD DISP=SHR,DSN=my.iam.file1  
//FILE2 DD DISP=SHR,DSN=my.iam.file2  
//FILE3 DD DISP=SHR,DSN=my.iam.file3  
//SYSIN DD *
   DIAGNOSE FROMDD=FILE1
   DIAGNOSE FROMDD=FILE2
   DIAGNOSE FROMDD=FILE3
/*
```

*Figure 10-113: Example of Diagnosing multiple files (EX1087B)*

One of the important considerations in recovering an IAM dataset is that the corrupted dataset may be needed for problem determination and resolution. If it is at all possible, it is best to save the existing problem dataset where it is, and recover into a new dataset. If that is not possible, then the next best choice is to back up the dataset with FDR from Innovation Data Processing, or a comparable software product, such as DFSMSdss from IBM. Refer to the section on backing up IAM datasets for information and examples of how to obtain a backup copy with one of those products. Other types of backup copies may not preserve the exact image of the dataset, which can result in not being able to perform problem determination. An IDCAMS REPRO, or the output dataset from IAMRECVR RECOVER command will not be adequate for performing problem resolution.
Equally important is to obtain some diagnostic information. If you call for assistance due to a potentially damaged file, you will be asked to obtain diagnostic information. The IAMRECVR PRINT command is used to help obtain some of this information. It will be quite helpful to run the job below prior to calling for technical assistance, as this is generally a diagnosis starting point. This job uses the PRINT command of IAMRECVR to print out the blocks within the dataset that describe the characteristics and physical structure of the file. A LISTCAT ALL is also being performed, as that will print out additional information on the volume(s) that the dataset resides.

```
//PRINTIDP  EXEC  PGM=IAMRECVR,REGION=4096K
//SYSPRINT DD   SYSOUT=*           
//DISKIN   DD   DSN=my.iam.dataset,DISP=SHR 
//SYSIN    DD   *                   
PRINT    IDPINC
/*
//LISTCAT  EXEC  PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*           
//IAMPRINT DD   SYSOUT=*           
//SYSIN    DD   *                   
LISTCAT ENT(my.iam.dataset) ALL
/*
```

Figure 10-114: PRINT IDPINC Example (EX1087C)

With information from the above PRINT or DIAGNOSE, you may also be asked to print out selected blocks from the IAM dataset. The IAM Technical Support representative will inform you if this is needed, and provide you with the block range(s) to be printed. This is accomplished with a different flavor of the PRINT command. In this case, a starting block number is specified by the FBLK= (from block) keyword, and the number of blocks to print is specified with the MAXBLKS= keyword. You can specify multiple PRINT commands in the same execution of IAMRECVR, each of which will specify different ranges of blocks to be printed. The example below shows how this is done.

```
//PRINTBLK EXEC  PGM=IAMRECVR,REGION=4096K
//SYSPRINT DD   SYSOUT=*           
//DISKIN   DD   DSN=my.iam.dataset,DISP=SHR 
//SYSIN    DD   *                   
PRINT   FBLK=65,MAXBLKS=4 
PRINT   FBLK=100,MAXBLKS=1
/*
```

Figure 10-115: Example of printing out blocks of an IAM Dataset (EX1087D)
Recovering IAM Datasets

**RECOVER**

To perform the recovery, the IAMRECVR RECOVER command is used to obtain a sequential copy of the data. If there are records in the Extended Overflow (or Independent Overflow for Compatible format files), the sequential output file must be sorted. The RECOVER command will invoke the sort product you have installed at your installation. The JCL you provide must specify whatever DD cards are needed for the sort. Frequently this requires specification of sort work space, with three or more SORTWK0x DD cards. This sort work space must be adequate enough to handle the amount of data that is contained within the dataset. The recover step is followed by a step to define a new cluster and reload the data with IDCAMS REPRO. The reload step also includes renaming the datasets.

**Spanned Records**

To perform a recovery on files with spanned records, there are additional considerations. On the step executing the IAMRECVR RECOVER command, an additional DD statement is required. That DD statement is SPANOUT. This DD statement defines a file on tape or disk that will contain those records that are actually spanned that is too large to fit within a single block. After performing the reload of the dataset from the TAPEOUT file an additional step is required for the spanned records. That step is to run the IAMRECVR APPLY command with the SPANNED keyword. That will update the recover file with the spanned records from the SPANOUT file. Further information on using these keywords is in Section 45.

**Example E: Basic Recover**

The example below performs both the RECOVER step and the REPRO step. First, the RECOVER command of IAMRECVR is executed to create a sequential file containing all of the data records from the IAM dataset. If that process is successful, then IDCAMS is executed. With IDCAMS a new dataset is defined, using the original dataset as a model, and then loaded with the recovered data. This is followed by renames of the datasets. Note the use of the IDCAMS IF and CANCEL commands which are done to preserve the original dataset in case a failure occurs during the IDCAMS processing.

For the execution of the IAMRECVR program, the DISKIN DD specifies the IAM dataset, and the TAPEOUT DD specifies the new sequential dataset. The SYSPRINT and SYSIN DD cards are required by IAMRECVR. The SORTWK0x and SYSOUT DD statements are provided for the SORT.

For the execution of the IDCAMS step, the SYSPRINT and SYSIN DD are required. The sequential file created by the RECOVER process in the prior step is included with a DD name of INFILE. The IAMINFO DD is optional, but recommended to obtain the run time report for the file load. Note that there is a DD statement, OLDIAMDS, for the original IAM dataset that is otherwise not referenced. This is done to hold the ENQ on the original dataset name until the recovery process is complete. Proper caution should be used in constructing the job stream to insure that the data is preserved.
Duplicate Keys

One of the circumstances that may occur is that records with duplicate keys are discovered by the RECOVER process after the sort has been done, while the output sequential dataset is being written. This circumstance does not necessarily represent a data integrity problem with the file. When a record is updated, the length of the record may be changed either by the application program itself, or by IAM if the updated record compresses differently. If there was an increase in the record length as a result of the update, the record may no longer fit within the current block that it resides, so it is moved by IAM to Overflow. Without Variable Overflow because the maximum length is reserved for a record once it is moved to an overflow block it will stay in that block. With Variable Overflow, the record may need to be moved to a different overflow block. IAM will first write out the updated record within the block it was moved to, and then
subsequently write out the original block with the old record deleted. If a failure occurs that prevents the proper closing of the dataset, the second write might not yet have been done resulting with the record existing in both blocks. Failures that may result in this condition include z/OS failures resulting in an IPL without proper application shutdown, using the z/OS FORCE command to cancel an updating job from the system, or other types of address space failures, or power outage. Files that were opened for update during such a failure should be reorganized or recovered as soon as possible after such a failure. Unfortunately, such failures also prevent the file statistics from being updated as well, so accurate information may not be reflected in the statistics particularly for the actual record count.

Other possibilities for duplicate keys include sharing the IAM dataset for concurrent updating or software failures that caused storage corruption. For these types of duplicates, you may need to examine which of the duplicate records you want to have in the recovered dataset. This can be accomplished by editing the LOG dataset that is created by IAMRECVR, and then running the APPLY step.

During normal IAM processing, the first duplicate record condition is not a problem as long as the record is not deleted. This is because with Enhanced format files, the record will always be moved to a higher relative block than it existed in before the update. So, the valid record will always be the record in the highest block. (Note that for Compatible format files, the situation is reversed because the Overflow area is at the physical beginning of the dataset.) For a recovery using IAMRECVR, a different procedure than the basic one shown in the preceding example must be used.

**Example F: Recover with Duplicate keys**

The first change to the original example is that the SORT must be told to pass records with equal keys back in the same order that they were passed to the SORT. This is done with the EQUALS option for DFSORT and SYNCSORT. For DFSORT, this option is specified by a control card input using the DFSPARM DD. For SYNCSORT, a $ORTPARM DD card is used. In the example, both are included. The next change for Enhanced format files is to specify a LOG dataset and indicating on the RECOVER command that records with duplicate keys are to be logged, (i.e. specify DUP=LOG on the RECOVER command). The first record of any specific key value will always be written to the normal sequential output dataset. Any subsequent records with the same key will be written out to the LOG dataset. Then, the normal reload is done for the dataset.

Then, for Enhanced format files only, the records in the LOG dataset are copied into the IAM dataset with an IAMRECVR APPLY statement. Note that you could also use an IDCAMS with the REPRO REPLACE statement. See Example G for using the REPRO REPLACE instead of the IAMRECVR APPLY. The advantage of using the IAMRECVR APPLY is that it will print out the keys of the records that are being replaced by the apply operation.
Figure 10-117: Example of Recovering Dataset With Duplicate Keys (EX1087F)

//RECOVER EXEC PGM=IAMRECVR,REGION=4096K
//SYSPRINT DD SYSOUT=* 
//SYSPRT DD SYSOUT=* 
//DISKIN DD DISP=OLD,DSNAME=my.iam.dataset
//TAPEOUT DD DSN=my.seq.dataset,DISP=(,CATLG),
//UNIT=SYSDA,SPACE=(CYL,(20,10))
//LOG DD DSN=my.duprec.dataset,DISP=(,CATLG),
//UNIT=SYSDA,SPACE=(CYL,(2,1))
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTPARM DD * 
EQLAUS
/*
//DFSPARM DD * 
EQLAUS
/*
//SYSIN DD *
RECOVER DUP=LOG
/*
//LOADNEW EXEC PGM=IDCAMS,COND=(0,NE)
//SYSPRINT DD SYSOUT=* 
//IAMINFO DD SYSOUT=* 
//INFILE DD DSN=my.seq.dataset,DISP=OLD
//SYSIN DD *
DELETE my.iam.dataset
IF MAXCC NE 0 THEN CANCEL
DEFINE CLUSTER(NAME(my.iam.dataset) - 
OWNER($IAM) - 
VOL(myvol) CYL(20 10) - 
RECORDSIZE(300 1000) - 
KEYS(16 0) - 
FREESPAC(10 10) - 
SHAREOPTIONS(2 3))
IF MAXCC NE 0 THEN CANCEL
REPRO INFILE(INFILE) ODS(my.iam.dataset)
LISTCAT ENT(my.iam.dataset) ALL
/*
//APPLY EXEC PGM=IAMRECVR,REGION=0M
//SYSPRINT DD SYSOUT=* 
//IAMINFO DD SYSOUT=* 
//LOG DD DSN=my.duprec.dataset,DISP=OLD
//VSAMOUT DD DSN=my.iam.dataset,DISP=OLD
//SYSIN DD *
APPLY OUT=VSAM
/*
In this next example, rather than deleting and redefining the dataset, it is copied into with a REPRO REUSE. If that is successful, then the duplicates, if any, are copied into the IAM dataset with another REPRO, but this time with REPLACE.

```bash
//RECOVER  EXEC PGM=IAMRECVR,REGION=4096K
//SYSPRINT DD   SYSOUT=*  
//SYSOUT  DD   SYSOUT=*  
//DISKIN   DD   DISP=OLD,DSNAME=my.iam.dataset  
//TAPEOUT  DD   DSN=my.seq.dataset,DISP=(,CATLG),  
//           UNIT=SYSDA,SPACE=(CYL,(20,10))  
//LOG      DD   DSN=my.duprec.dataset,DISP=(,CATLG),  
//           UNIT=SYSDA,SPACE=(CYL,(2,1))  
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))  
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))  
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))  
//$ORTPARM DD   *  
//DFSPARM  DD   *  
//SYSIN    DD   *  
RECOVER  DUP=LOG  
/*  
//LOADNEW  EXEC PGM=IDCAMS,COND=(0,NE)  
//SYSPRINT DD   SYSOUT=*  
//IAMINFO  DD   SYSOUT=*  
//IAMFILE  DD   DSN=my.iam.dataset,DISP=OLD  
//INFILE   DD   DSN=my.seq.dataset,DISP=OLD  
//DUPFILE  DD   DSN=my.duprec.dataset,DISP=OLD  
//SYSIN    DD   *  
  REPRO INFILE(INFILE) OUTFILE(IAMFILE) REUSE  
IF MAXCC NE 0 CANCEL  
REPRO INFILE(DUPFILE) OUTFILE(IAMFILE) REPLACE  
LISTCAT ENT(my.iam.dataset) ALL  
/*
```

Figure 10-118: Example of recover with REPRO (EX1087G)
10.88 IAM Journal and Recovery

This section provides information on using IAM journaling and recovery for IAM datasets not processed by IAM/PLEX or IAM/RLS. For the journaling and recovery offered for datasets processed with IAM/RLS, refer to Section 20 on Using IAM/RLS or Section 25 on Using IAM/PLEX.

Overview

For Enhanced Format IAM datasets without spanned records, IAM offers an optional journaling facility along with an associated recovery capability. IAM journaling is activated through IAM Overrides for the datasets that you select. The IAM journaling facility provides a tool to capture the images of records prior to being updated, called before images, and/or images of records after being updated, called after images. Once a journal has been created, it can be used by the provided utility program IAMJREST to recover the IAM dataset to a particular point in time. There are two types of recovery possible, called forward recovery and backward (or backout) recovery. Forward recovery is done by restoring a backup copy of the affected IAM dataset, then applying the after images from the journal dataset(s) to the IAM dataset. In other words, the recovery proceeds forward from a particular point in time to an identified point in time, prior to the failure. A backout recovery does the opposite function. Starting with a dataset that has been updated by various jobs, it provides a capability to backout the updates from selected jobs. This is accomplished by updating the dataset from the log using the before images.

There are two choices for journaling datasets. The first method is that each IAM dataset using journaling will have a separate and unique journal dataset. The second choice is to use the IAM RLS journaling even if the IAM dataset being journaled is not using IAM RLS processing. The choice of which journaling method to use is taken from the IAM Global Options Table, option RLSJRNL. If that option is ENABLED then journaling will be handled by the IAM RLS address space. If the option is DISABLED, which is the default journaling is performed within the address space of the job(s) using the IAM dataset, to a unique journal dataset for the IAM file.

Why Use IAM Journaling

The main reason to use the IAM journaling feature is to improve data availability for IAM datasets. This is accomplished because the enhanced recovery capabilities can reduce the frequency of backing up the entire dataset and/or eliminate the need to restore IAM datasets when jobs abend or otherwise fail.

If the after images are journaled, then rather than backing up large IAM datasets every day, they can be backed up less frequently, perhaps just once a week. Only the journal containing the updated data would be backed up on a daily basis. By backing up less data, the amount of time a dataset is not available for update processing is reduced. The amount of time saved by not doing full backups each night can be quite substantial. In case of a media failure, the dataset is recovered by restoring the dataset from the last full backup, then executing a journal forward recovery using IAMJREST.
If before images are journaled then the time to recover after an abend or other failure can be reduced. This is accomplished by backing out the updates and then resuming processing. This avoids having to restore the entire contents of data sets, and potentially rerunning several jobs. The updates made by the failing batch job can be backed out by IAMJREST, and then the batch job processing can be restarted. If needed, the updates from multiple jobs steps and jobs can be backed out. The backout facility may be able to save a lot of time when recovering from typical job abends, by eliminating the need to restore datasets and rerun an entire sequence of batch jobs. Additional savings are possible by eliminating the need for batch jobs to perform separate backup of critical files prior to starting an update process.

There are some distinct advantages of using IAM RLS journaling. First, the journal datasets can be managed at an installation level, rather than managing what could be many individual journals at an application level. Second, if jobs are updating multiple IAM datasets, then only one execution of the recovery program (IAMJREST) is needed to perform the recovery on all of the journaled IAM datasets. Likewise, if some of the IAM datasets are using IAM RLS processing, and some IAM datasets are not using IAM RLS, there is only one execution of the recovery program needed, and the automatic backout function of IAM RLS can be used. Additionally long running jobs that are updating IAM datasets can make use of the batch syncpoint capabilities when using IAM RLS journaling, which can further reduce recovery and restart times.

Preparing for Journaling

Before turning on IAM journaling for a dataset you need to take a few steps in preparation. First, you must decide what types of recoveries you are going to want to perform, as this choice impacts the type of records you will need to have written to the journal dataset. If you are planning on both the forward and backward recoveries, you will need to be collecting both before and after images. If you are planning on just forward recoveries, then only the after image records are collected. Conversely, if you are planning on only backout recoveries you only need to collect before images. You can also choose between journaling all update activity against a file, or only journaling for some selected jobs. If you are journaling for all or most of the update activity on the IAM dataset, then you should activate journaling by specifying the IAM CREATE JRNAD= override when the file is defined or loaded. The journaling option you select will be retained with the dataset, and used on each access to the dataset. If you are only planning on using the journaling for a few selected jobs to be able to backout their updates, then the journaling can be specified on the selected jobs steps using the IAM ACCESS JRNAD= override. As operands on the JRNAD keyword, you specify the types of records that you need to have written to the journal. Valid values are:

- BEFORE for before images,
- AFTER for only updated record images, or
- BOTH to collect BEFORE and AFTER images.

Using BOTH will provide the most flexibility, but also incurs the largest overhead in terms of space required and journal I/O activity. When estimating the space, keep in mind that IAM does NOT journal during file load or reorganization, so there will not be a “second copy” in the journal of the entire dataset.

RLS Journaling

If your installation is using the IAM RLS journaling capabilities, as indicated by setting the IAM Global Option ENABLE=RLSJRNL, information on using those journals is provided in Section 20.30, Section 20.60, and Section 20.70 of the IAM Manual. These sections include information on setting up the IAM RLS journals, managing the IAM RLS journals, batch syncpoint facility, and recovery using the IAM RLS journals.
Managing the Journal Dataset

If your installation is NOT using IAM RLS journaling, as indicated by defaulting or setting the IAM Global Option DISABLE=RLSJRNL, then you are responsible for allocating and managing the journal dataset(s).

When using IAM journaling for files that are used under CICS, the files MUST BE IN AN LSR POOL.

The next step is to decide how you want to manage your IAM journal datasets. The IAM Journal datasets must reside on DASD while they are being actively used for journaling. They can reside on either tape or disk when performing the recovery phase. If you are opting for the full journaling, then you will need permanent DASD datasets for the journals. Typically, the journal datasets will be backed up daily using IEBGENER. The reason for using IEBGENER is so that the backup copy of the LOG dataset can be used directly as input to the IAMJREST utility. That would not be possible if the dataset was backed up by DASD management software products, such as FDR/ABR. The backup copy can take the form of an accumulation dataset or as a separate dataset each day, perhaps by using a GDG.

To help minimize both the amount of DASD space for the journal and the journal backup time, you will probably want to empty the journal dataset(s) after they are backed up. This can be easily accomplished by running program IEBGENER with an empty input dataset. Or, you might decide that it is more critical to minimize the time it takes to set up the recovery. In that case you would want to accumulate the journal data within the journal dataset itself, containing all the data since the last full dataset backup.

If you are just going to do journaling for batch update jobs to provide a backout recovery capability, you have some different choices. You can either go with a permanently allocated journal dataset, or allocate one at the beginning of the batch job stream. In either case, you will most likely want to backup the journal dataset at the end of the batch stream. If you are using a permanent journal dataset, you will want to empty it prior to or at the beginning of the job stream that is being journaled.

Estimating Journal DASD Space

With the above decisions made, you can next estimate the amount of DASD space that will be needed for your journal dataset. To make the estimate, you will need some statistical information that can be found in IAM LISTCAT or IAMINFO reports. These reports should cover the typical length of time that the data will be residing in the journal, whether that is just for a day, a week, or through a batch job stream. The important numbers are the number of inserts updates and deletes for the length of time in question, as this will be used to calculate the number of records that are written to the journal. To estimate the number of records, which we will call R, select one of the following calculations based on the types of records that are being collected:

- For both BEFORE and AFTER Images:  \( R = (2 \times (\text{updates} + \text{deletes})) + \text{inserts} \).
- For only BEFORE or AFTER Images:  \( R = \text{updates+deletes+inserts} \).
The IAM journal will typically use ½ track blocking, which will be 27998 when residing on a 3390 and using variable length records and blocks. The maximum record length is the defined user record size plus 52 bytes for the header information on each journal record. If the maximum record size exceeds the ½ track block size, then a block size of 32760 will be used. To play it absolutely safe, you should estimate your journal space requirements using the maximum record length. However, for files that have a very large maximum record length, for which you know that the actual average record length is considerably smaller, you can use the smaller record length. Select the record size that seems most appropriate, and add 52 bytes to that length for the header information. Divide the expected block size minus four by the record size to get the average number of journal records per block, dropping any fraction. Multiply the result by 30 if using ½ track blocking, or by 15 if the larger block size of 32760 must be used. That will yield the number of journal records per cylinder, which we will call C.

The estimated number of cylinders will be R / C, which is to be rounded up. Or in words, the estimated number of log records divided by the number of records per cylinder yields the number of cylinders. Use this value for the primary space allocation of the journal dataset. Then use some fraction in the range of 10% to 25% as the secondary space value for your journal dataset allocation.

Example of Journal Space Calculation

As an example, let's say that we have a file that typically has 15,000 updates, 1,000 inserts, and 500 deletes in a day. Our plan is to collect both before and after images, enabling both forward and backout recovery. The journal will be backed up and then emptied each day after the online system comes down, prior to starting the batch update runs. The estimated number of daily journal records will be:

- \( R = 2 \times (15,000 + 500) + 1,000 = 32,000. \)

The maximum defined user record length is 256, so the maximum journal record length is (256 + 52) = 308. The journal is going to reside on a 3390, which has a ½ block size of 27,998. Dividing (27,998-4) by 308 = 90 log records per block. Next, multiply the number of log records per block times the number of blocks per cylinder (which is 30), to come up with 2,790 log records per cylinder.

- \( C = \frac{(27998-4)}{(256 + 52)} \times 30 = 2,700. \)

The number of cylinders required is \( R / C \) or \( 32,000 / 2,700 = 12. \)

Journal Dataset Name

The name of the journal dataset is the dataset name (or cluster name) of the IAM dataset, with the characters `.LOG` appended to the end. If the length of the dataset name exceeds 40 bytes, then the end will be overlaid with the `.LOG` character string and be 44 bytes long, except when the 40th byte is a `.`, In that case, the dataset name will only be 43 bytes long, still ending with the `.LOG` literal.

Journal Allocation Concerns

There are a few considerations for doing the actual journal allocation. First, to maximize the recovery potential, and to avoid I/O contention between the journal and the actual dataset, the journal dataset must be on a different volume. It would be best for the journal dataset to be on a device that is on a different channel and controller if that is possible. Secondly, make sure to pick a volume that has sufficient DASD space available to allow the journal dataset to go into extents if necessary. Thirdly, do not specify any DCB characteristics, such as LRECL, BLKSIZE, or RECFM. The IAM journal program will set those values automatically the first time the journal dataset is used.
The example JCL below demonstrates defining an IAM dataset with the journaling option requested on the IAM CREATE Override statement. Note that JRNAD=BOTH is specified, indicating journaling of both before and after images. This option provides for both forward and backout recoveries. The journal dataset is also being allocated in the same job step, using the SPACE value calculated in the above example calculation.

```
//IAMDEFIN EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//JOURNAL DD DSN=MY.IAM.KSD.LOG,  
//          UNIT=SYSDA,DISP=(,CATLG),  
//          SPACE=(CYL,(12,6)),VOL=SER=MYVOL2  
//IAMOVRI DDD *  
//CREA T DD=&ALLDD,JRNAD=BOTH  
/*  
//SY SIN DD *  
   DEFINE CLUSTER -  
   (NAME(MY.IAM.KSD) -  
   OWNER($IAM) -  
   VOLUMES(MYVOL1) -  
   CYL(60 6) -  
   RECORDSIZE(100 256) -  
   KEYS(24 8) -  
   FREESPAC E(5 20) -  
   SHAREOPTIONS(2 3) -  
   REUSE )  
   LISTCAT ENT(MY.IAM.KSD) ALL  
/*
```

Figure 10-119: Example of Defining an IAM Dataset with Journaling (EX1088A)
Example B: Setting up GDG for Journal Backups

To continue with the above example, the next step is setting up whatever JCL is needed for management of the log data. For the IAM dataset being journaled, the backup frequency had been once a day. With the IAM journal being active for this dataset, the backup frequency will be changed to once a week. The journal will be backed up daily, to a GDG, and then emptied. As part of the weekly dataset backup, we will accumulate the daily journals for the past week, into a single dataset on tape, which will be the weekly journal GDG. Step 2 is then to set up the GDG for the journal backups, which can be done as shown below:

```plaintext
//IAMDEFIN EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
   DEFINE GENERATIONDATAGROUP -
      (NAME(MY.IAM.KSD.DAILY.JOURNAL) -
         LIMIT(?) SCRATCH )
   DEFINE GENERATIONDATAGROUP -
      NAME(MY.IAM.KSD.WEEKLY.JOURNAL) -
         LIMIT(54))
/*
```

Figure 10-120: Defining GDG for Journal Backups (EX1088B)

Example C: Daily Journal Backup Job

The daily journal backup job will run six days a week. This job will not only make a copy of the journal dataset to the new daily journal backup GDG, but will also empty the journal file so that the most it will contain is one day’s worth of data. The following example will show how this can be accomplished:

```plaintext
//BKUPJRNL EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=* 
//SYSIN DD DUMMY 
//SYSUT2 DD DSN=MY.IAM.KSD.DAILY.JOURNAL(+1),
//      DISP=(,CATLG),SPACE=(CYL,(12,6)),UNIT=SYSDA,
//      DCB=(MY.IAM.KSD.LOG),VOL=SER=MYVOL3
//SYSUT1 DD DSN=MY.IAM.KSD.LOG,DISP=OLD
//EMPTYJRN EXEC PGM=IEBGENER,COND=(0,NE,BKUPJRNL)
//SYSPRINT DD SYSOUT=* 
//SYSIN DD DUMMY 
//SYSUT1 DD DUMMY,DCB=(MY.IAM.KSD.LOG)
//SYSUT2 DD DSN=MY.IAM.KSD.LOG,DISP=OLD
```

Figure 10-121: Example of Daily Journal Backup (EX1088C)
Example D: Weekly Journal Backup

The weekly journal backup job should be run as part of the same job that performs the backup of the base IAM dataset. Preferably as the step immediately after the dataset backup. If time is critical, the weekly backup of the journal datasets can be done concurrent with the dataset backup. This job will copy the previous six daily backups along with any current data in the journal, and subsequently empty the journal dataset itself.

```plaintext
//WBKUPJRN EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=* 
//SYSIN DD DUMMY
//SYSUT2 DD DSN=MY.IAM.KSD.WEEKLY.JOURNAL(+1),
//       DISP=(,CATLG),UNIT=CART
//SYSUT1 DD DSN=MY.IAM.KSD.DAILY.JOURNAL(-5),DISP=OLD
//       DD DSN=MY.IAM.KSD.DAILY.JOURNAL(-4),DISP=OLD
//       DD DSN=MY.IAM.KSD.DAILY.JOURNAL(-3),DISP=OLD
//       DD DSN=MY.IAM.KSD.DAILY.JOURNAL(-2),DISP=OLD
//       DD DSN=MY.IAM.KSD.DAILY.JOURNAL(-1),DISP=OLD
//       DD DSN=MY.IAM.KSD.DAILY.JOURNAL(0),DISP=OLD
//       DD DSN=MY.IAM.KSD.LOG,DISP=OLD
//EMPTYJRN EXEC PGM=IEBGENER,COND=(0,NE,WBKUPJRN)
//SYSPRINT DD SYSOUT=* 
//SYSIN DD DUMMY
//SYSUT1 DD DUMMY,DCB=(MY.IAM.KSD.LOG)
//SYSUT2 DD DSN=MY.IAM.KSD.LOG,DISP=OLD
```

Figure 10-122: Weekly Backup to Tape of Journal Data (EX1088D)

Journal Recovery Procedures

Now that the journaling is all in place, along with the proper backup procedures, the recovery process from the IAM journal can be examined. Recovery with the journal is provided for by the IAMJREST program. The IAMJREST program uses as input the journal dataset(s), the base IAM dataset, and some control parameters directing its operation. The output of IAMJREST is the recovered IAM dataset. This section will provide the basics of using IAMJREST, along with examples. For additional information in IAMJREST, refer to Section 47 of the manual, which has a complete description of the IAMJREST program.

Forward Recovery

There are two different types of recovery procedures that can be performed by IAMJREST. The first type is a forward recovery. A forward recovery will typically be used to recover a dataset if the device it resides on is damaged or not operational, or if the dataset has been seriously corrupted. A forward recovery is one where first the base dataset is restored, then the IAMJREST utility is executed to apply the updates, inserts, and deletes from the journal to the base dataset. For a forward recovery to occur, the journal must contain AFTER images for all the jobs that updated the dataset. This is accomplished by specifying JRNAD=BOTH or JRNAD=AFTER on the IAM CREATE override statement.
Example E: Forward Recovery with Restore

Shown below is an example of performing a forward recovery. In the example, it is presumed that a failure occurred on the third day after the last backup. So, there are the two daily backups of the journal file, plus the current journal file that will be used for the recovery. These three datasets will be input to the recovery procedure. The dataset being recovered will first be deleted and defined on a new volume, then it will be reloaded from the backup tape. After the restore, IAMJREST is executed with the three input files to perform the forward recovery. The failure occurred while running the job BATCHUP2. The recovery will be performed for all update jobs that ran up to and including job BATCHUP1. Once any other datasets are recovered, the BATCHUP2 job can be restarted.

```
//DELDDEFIN EXEC PGM=IDCAMS
//SYSPRINT DD SYSPUT=*
//IAMOVRDD *
CREATE DD=&ALLDD,JRNAD=BOTH
/*
//SYIN DD *
DELETE MY.IAM.KSD CLUSTER
IF MAXCC NE 0 THEN -
DELETE MY.IAM.KSD NOSCRATCH
SET MAXCC = 0
END
DEFINE CLUSTER -
(NAME(MY.IAM.KSD) -
OWNER($IAM) -
VOLUMES(NEWVOL) -
CYL(60 6) -
RECORDSIZE(100 256) -
KEYS(24 8) -
FREESPACE(5 20) -
SHAREOPTIONS(2 3) -
REUSE )
LISTCAT ENT(MY.IAM.KSD) ALL
/*
//RESTORE EXEC PGM=IDCAMS,COND=(0,NE)
//SYSPRINT DD SYSPUT=*
//BACKUP DD DSN=MY.IAM.KSD.BACKUP,DISP=OLD
//IAMFILE DD DSN=MY.IAM.KSD,DISP=OLD
//IAMINFO DD SYSPUT=*
//SYINDD*
REPRO INFILE(BACKUP) OUTFILE(IAMFILE) REUSE
LISTCAT ENT(MY.IAM.KSD) ALL
/*
//RECOVER EXEC PGM=IAMJREST,REGION=64M,COND=(0,NE)
//SYSPRINT DD SYSPUT=* 
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50,10))
//IAMJRNLD DSN=MY.IAM.KSD.DAILY.JOURNAL(-1),DISP=OLD
// DD DSN=MY.IAM.KSD.DAILY.JOURNAL(0),DISP=OLD
// DSN=MY.IAM.KSD.LOG,DISP=OLD
//IAMINFO DD SYSPUT=* 
//SYINDD*
RESTORE FORWARD,TOJOB=BATCHUP1,TODATE=2009182
/*
```

Figure 10-123: Forward Recovery Preceded by a Restore (EX1088E)
The other type of recovery that can be performed is a backout (or backwards) recovery. This type of recovery backs out updates to a dataset from the specified jobs or job steps. To perform a backout recovery, the journal must include BEFORE images. BEFORE images are written to the journal when either JRNAD=BOTH or JRNAD=BEFORE have been specified on the IAM overrides. A forward recovery can be done to restore a dataset to the same point that can be done with a backout recovery. The backout recovery however will generally be much faster than a forward recovery. This is because the dataset does not have to be restored, and eliminates having to apply updates from potentially several days. Backout recoveries are also ideal if you only need to use IAM journaling to provide recovery for a few jobs or a job stream.

**Example F: Backout Recovery**

Shown below is an example of a backout recovery. In this recovery, all the updates for job BATCHUPD that was run on June 30, 2009 (2009.181) are removed from the IAM dataset.

```plaintext
//RECOVER EXEC PGM=IAMJREST,REGION=64M,COND=(0,NE)
//SYSPRINT DD   SYSOUT=* 
//SYSOUT DD   SYSOUT=* 
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(50,10))
//IAMFILE DD   DSN=MY.IAM.KSD,DISP=OLD
//IAMJRNL DD   DSN=MY.IAM.KSD.DAILY.JOURNAL(-1),DISP=OLD
//   DD   DSN=MY.IAM.KSD.DAILY.JOURNAL(0),DISP=OLD
//   DD   DSN=MY.IAM.KSD.LOG,DISP=OLD
//IAMINFO DD   SYSOUT=* 
//SYSIN DD   *
RESTORE BACKOUT,JOBNAME=BATCHUPD,FROMDATE=2009181
/*
```

*Figure 10-124: An Example of a Backout Recovery (EX1088F)*

**Example G: Identifying the Contents of a Journal**

If you are unsure of what jobs or job steps have been journaled, the IAMJUTIL provides the SCAN capability. This command will read the specified journal dataset, and print a line identifying each of the job steps that have records on the file. Included with that information is the time and date. This information may be helpful when setting up to run a recovery. The SCAN command also has a DETAIL operand, which provides a way to determine the complete contents of the journal dataset, should that be needed. Normally though, it is expected that the summary information identifying all the job steps that have updated the dataset will be sufficient information.

```plaintext
//RECOVER EXEC PGM=IAMJUTIL
//SYSPRINT DD   SYSOUT=* 
//IAMJRNL DD   DSN=MY.IAM.KSD.LOG,DISP=OLD
//SYSIN DD   *
SCAN SUMMARY
/*
```

*Figure 10-125: Example of Identifying Contents of Journal (EX1088G)*
Adding Volumes to an IAM Dataset

Users can add volume(s) to their IAM datasets through the IDCAMS utility. One of two different procedures will need to be followed depending on whether or not the IAM dataset is SMS managed. Some basic guidelines are to be followed regardless of the procedure used.

1. While not required, it is highly recommended that the dataset not be allocated to any other job at the time that this process is done. This recommendation is particularly true if the system being used has a software product that can dynamically add a volume should an X37 type of abend occurred while it is being used, and could occur during or after the add volume process. The catalog entry for the dataset could become unusable should this occur. To enforce this guideline it is recommended that a DD card for the dataset be included in the job that is used to add the volume with a DISP=OLD to ensure exclusive use.

2. If the above guideline was not followed, then after adding the volume(s) any job or other address space that has the dataset allocated and possibly open will need to close and de-allocate the dataset. Upon reallocating the dataset the new volume(s) will be available to those address spaces.

Example A: Adding a volume to an SMS Managed Dataset

To add a volume to an SMS managed IAM dataset, use the IDCAMS ALTER ADDVOLUME request, as shown in the example below:

```
//EX1089A  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=* 
//IAMFILE  DD DSN=IAMV.EX1089A.CLUSTER,DISP=OLD
//IAMPRINT DD SYSOUT=* 
//SYSIN    DD * 
   ALTER IAMV.EX1089A.CLUSTER ADDVOLUMES(*) 
   LISTC  ENTRIES(IAMV.EX1089A.CLUSTER) ALL
/*
```

Figure 10-126: Example of adding a volume to an SMS managed IAM dataset (EX1089A)
Adding a volume to a non-SMS managed IAM dataset requires a bit more effort. First, you need to get a current listing of the catalog with the current volumes. Then you build the job that will add the volume, which does the following process:

1. Uncatalog the dataset
2. Recatalog the dataset with the updated volume list. Make sure that all prior volumes are included in their proper order.
3. List out the new catalog entry for manual verification.

An example of this process is shown below:

```
//EX1089B EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//IAMPRINT DD SYSOUT=* 
//IAMFILE DD DSN=IAMV.EX1089B.CLUSTER,DISP=OLD 
//SYSIN DD * 
DELETE IAMV.TEST.CLUSTER NOSCRATCH 
DEFINE CLUSTER(NAME(IAMV.EX1089B.CLUSTER) - 
   OWNER($IAM)           - 
   VOLUMES(EXISTINVOL NEWVOL) - 
   RECATALOG ) 
LISTC ENT(IAMV.EX1089B.CLUSTER) ALL  
/*
```

Figure 10-127: Adding a volume to a non-SMS Managed IAM dataset (: EX1089B)
20 USING IAM RECORD LEVEL SHARING

IAM Record Level Sharing

Overview

IAM Record Level Sharing (RLS) for a single system was introduced in IAM Version 8.0. With IAM/RLS concurrently running batch jobs, TSO users, and CICS regions running on the same LPAR can safely update the same IAM files, if those files are accessed through IAM/RLS. IAM is different from VSAM/RLS in that rather than having an application such as CICS indicate it is using IAM/RLS, IAM will automatically select those datasets that should be processed by IAM/RLS. Eligibility for processing under IAM/RLS is determined during OPEN processing, on a file-by-file basis. When an eligible IAM data set is opened, all of the I/O requests for that data set are passed to the IAM/RLS address space for processing, and the results returned to the requesting address space.

SYSPLEX Record Level Sharing is available in IAM with the IAM/Plex Record Level Sharing Support. Please see Section 25 of this manual for detailed documentation of IAM/PLEX. IAM/PLEX is an optional feature of IAM available for additional cost.

Version 9.2 Enhancements

Summary of changes for Version 9.2:

- Support for VSAM/RLS only I/O requests that may be issued from CICS application programs, including use of the operands Consistent, Repeatable, Nosuspend and Uncommitted. IAM/RLS is not currently able to provide support for the update browse operations due to the unique nature of the undocumented VSAM/RLS interfaces.
- Recognize and provide proper record locking function when the CICS parameter CILOCK=YES is being used with non-recoverable files.
- Special startup process that will activate the IAM VSAM Interface (VIF) and load the IAM/RLS and IAM/PLEX dataset name table without actually activating the IAM/PLEX address space. This is done by issuing the START command for the IAM/PLEX proc with PARM=DSNTLOAD. The IAM VSAM interface will be initialized and the dataset name table for IAM/PLEX will also be loaded into virtual storage. This will enable IAM to process data sets when the RLS Global Option indicates TABLE and REQUIRED before the IAM/PLEX address space is running.

IAM/RLS Capabilities

IAM/RLS provides the following capabilities:

- Ability to share IAM files for update with data integrity between concurrently executing batch jobs, TSO users, and CICS regions.
- Locking during update processing is performed at the record level, utilizing the IAM/RLS record locking facility. Record locks can be reestablished after a failure or shutdown of an IAM/RLS address space to facilitate recovery processing.
- For recoverable files, record locks are held until the job or transaction terminates, or the job or transaction issues a SYNCPOINT.
- Potential deadlocks within the resources managed by IAM/RLS are detected, and the associated request will be failed to prevent a deadly embrace.
- Journaling and recovery capabilities are provided for data sets processed under IAM/RLS and optionally for IAM datasets not processed under IAM/RLS. Journaling can be done either to sequential data sets, or to the z/OS System Logger.
• Ability to dynamically back out updates made to files processed by IAM/RLS when the batch job step that performed those updates abends.

• A callable batch syncpoint process is provided, to prevent batch programs from locking out access to large portions of recoverable files when mass updates are being performed.

• An AUTOSYNCPPOINT feature that will automatically issue syncpoints without changes to the application can be used to avoid changing batch application programs when syncpoints are required.

• An Automatic Disconnect and Reconnect capability within CICS is provided to dynamically disconnect and reconnect an IAM/RLS address space that has terminated and subsequently has been restarted. Any pending inflight updates by CICS transactions to IAM files processed by the IAM/RLS address space at the time of its termination will be scheduled for backout. However since the IAM files are not available, any backout to an unavailable IAM file will fail. CICS then marks such failed backouts as pending Shunted Unit-of-Work (UOWs). Subsequently, when the terminated IAM/RLS address space is available IAM will cause the shunted UOWs to be retried. Then IAM sets the files within CICS as being open and available for normal processing as IAM will have then fully reconnected to the restarted IAM/RLS address space. Additionally, the CICS region will not have to be restarted or recycled as in prior IAM releases.

• An ISPF interface to monitor activity within the IAM/RLS address space, and issue some of the operator commands.

• Depending on the RLS selection criteria implemented at an installation, there generally will be little or no JCL changes required to use IAM/RLS. For CICS, the installation and activation of certain exits is required, as documented in the IAM/RLS CICS Considerations in Section 20.50. For batch jobs that perform a large volume of updates to recoverable files being processed by IAM/RLS, the implementation of the IAM Batch Syncpoint, and appropriate restart capabilities are highly recommended.

**Record Locking**

For the single system record level sharing, IAM utilizes its own record level lock manager. Records are locked by their key within each IAM data set being processed by IAM/RLS. The IAM lock manager does check for potential deadlocks within the scope of the IAM data sets that it is managing. If waiting on the record lock for the current request will result in a deadlock, the request will be failed with a logical error. The trigger for IAM to release a record lock will depend on the environment in which the lock was requested.

For CICS transactions, records are locked by transaction and unit of work (UOW) identification. IAM determines from CICS which IAM files are recoverable, and will hold the record lock(s) until either a SYNCPPOINT is executed, or the transaction ends. If the IAM file is defined with the explicit option of JRNAD=NONE or the IDCAMS parameter LOG(NONE), then IAM will assume that the file is not recoverable under CICS. When an IAM file is detected as being not recoverable under CICS, then the record lock is only held from the time of the GET for UPDATE until the record is actually updated or erased, or for records being added, only for the duration of the actual add processing.
For other than CICS processing, that is batch jobs or TSO users, IAM will generally only hold the record lock(s) from the time of the GET for UPDATE until the record is actually updated or erased, or for records being added, only for the duration of the actual add processing. The exception to this is if IAMRLS is journaling before images of records, implying that a back out could be performed if there is a failure, then the record lock(s) will be held until the program either calls the IAM batch syncpoint, or the job step terminates. If a job step abends while processing a recoverable file, then any record locks obtained for that job step will be retained until a recovery takes place. If there were any other jobs or CICS transactions waiting for the record locks that are being retained until recovery, those requests will be failed when the abend has been detected. The recovery can be performed by IAM’s Dynamic Job Back Out function, by IAMJREST, or by whatever other procedures or recovery software that may be available. Information about any retained locks can be found by using the DISPLAY,RETAINEDLOCKS modify command to IAM/RLS. Such retained locks will be released upon successful recovery by IAMJREST or Dynamic Job Back Out. The retained locks can also be released by the RELEASELOCKS command.

IAM/RLS has an optional capability to reestablish record locks on restart of an IAM/RLS address space after either an unplanned or a planned outage. Only the record locks that are for recoverable IAM datasets will be reestablished. The record lock status for locks eligible for recovery are kept in a separate lock recovery dataset. To minimize the performance impact, the records will only be periodically written to disk after a specified lock check point time interval.

When the IAM/RLS address space is started, the record lock recovery process will read the lock recovery dataset to reestablish record locks. If the last status was not from a proper shut down then the IAM/RLS journals will be read from the time of the last lock checkpoint to update the record locks from the recorded activity. Once the record locks are reestablished, then the IAM/RLS address space is ready to resume processing.

In addition to handling the I/O and locking services, the IAM/RLS address space can also be used for journaling before and / or after images of updated records. The journaling can be performed either to standard DASD sequential data sets, or can be done using the z/OS System Logger. The IAM journaling services are primarily provided to allow for the back out of updates performed by failing batch job(s), or to perform a forward recovery of updates if a file has encountered media damage. CICS will handle its own transaction back out and other recovery as it does today, using its own logging mechanisms that is independent of the journaling provided by IAM. By default, IAM/RLS will not journal before images for records being updated by CICS transactions because that is done by CICS. If customers want IAM to journal the before images from CICS transactions, they can use the IAM/RLS startup parameter CICSJOURNAL.

Information and instructions on setting up IAM/RLS journaling is presented in Section 20.30.
Replication Logging

Support has been added to produce journal records to a System Logger logstream in the same format written by CICS in the CICS journals to support the use of replication software such as the Record Recovery Data Facility RRDF™ from E-Net. This logging facility is activated by adding the ALTSYSLG DD statement to the IAM/RLS or IAM/PLEX proc. The DD statement should specify the name of the alternate logstream to be used for these records. This is a 16 character name like, IAMLOGR.ALTGRP01. Then you have to specify the IAM ACCESS override ALTLOGGER for the datasets that you want these alternate journal records written for. The ACCESS override would have to be in the overrides used by the IAM/RLS or IAM/PLEX address space, and System Logger must also be active for the normal IAM/RLS or IAM/PLEX journals. Only after images will be written to this log which are adds, updates, and deletes.

Dynamic Job Back Out

An optional Dynamic Job Back Out function for IAM files opened through IAM/RLS is available. Whenever a job step abends, all updates done by that job step can be automatically removed by Dynamic Job Backout. Control of Dynamic Job Back Out is provided through IAM/RLS DJB startup parameter. If permitted by the startup parameter, DJB can also be controlled via an IAM ACCESS Override. If the batch job step or TSO user has taken IAM syncpoints, then the backout will be performed to the most recent syncpoint taken by the job step prior to abending. If no IAM syncpoints have been taken, then all of the updates performed by the failing job step will be removed on files accessed through IAM/RLS. Upon successful completion of the Dynamic Job Backout, all of the retained locks for the failing job will be released.

Security

The IAM/RLS address space will have to be given security authority to update those files that are going to be processed by IAM record level sharing. This will include CREATE authority for files that are can be extended to additional volumes. IAM does issue the RACROUTE macro within the individual job’s address space to validate that the requesting user does have authority to read or update, as appropriate, the IAM data set being opened prior to requesting that IAM/RLS open the data set. If the RACROUTE indicates that the user / job does not have authority to access the data set, then the OPEN request is failed. The failing job will receive an IAMW18 error message. Additional security will be performed by z/OS services when IAM attempts to take an extent

Reliability

IAM/RLS utilizes the various z/OS error handling and recovery facilities to recover from errors and abend conditions that may occur. IAM goals in error recovery routines are to provide continuous availability of the services being provided by the IAM/RLS address space, and to automatically collect enough information about any failures such that problem determination and correction can be performed from the single failure. The job that had submitted an I/O request to IAM/RLS that has experienced a failure will generally be failed with a VSAM logical error code. It will be up to the program to decide whether it can continue processing.
The error data collection will include various messages to the job log, system log, and to the RLS log indicating the abend code, general registers, access registers if applicable, and the failing module. The error routines do attempt to not duplicate diagnostic information that is produced by z/OS, but rather provide additional diagnostic information to be combined with the information provided by z/OS. The error information contained in these messages may be sufficient for problem determination and resolution, particularly if a problem had been previously reported to Innovation. If the error occurred within the IAM/RLS address space, an error trace table will be kept in storage for reference, particularly in situations where multiple errors have occurred. Most error situations will also result in a request for a dump to be taken to a system dump data set, which may include both the IAM/RLS address space and possibly the address space of the job that submitted the failing request. The IAM index space associated with the IAM/RLS address space may also be dumped.

Serviceability
To aid in serviceability a mechanism is provided for system support personnel to apply critical fixes to the IAM modules within the IAM/RLS address space, without the need to shut down the address space. This facility can also be used to backout fixes should they cause other problems. Refer to the IAM/RLS operator commands APPLY and RESTORE in Section 20.40.

HiperDispatch Enhancements
Some performance issues were discovered when running with HiperDispatch mode enabled on LPARs that had more than 4 engines assigned when doing heavy batch processing. The key problem areas were determined to be local lock contention, and dispatching overhead that was impacted by the lock contention. For most I/O requests, IAM used fewer cycles than it was taking to dispatch the work. Based on that information a number of enhancements have been made to the IAM/RLS processing to alleviate those issues, which include the following:

- Implemented use of the Perform Locked Operation (PLO) instruction when updating various queues that had been serialized with the local lock.
- Revised key paths to eliminate use of various SVC services that could serialize on various z/OS locks including GETMAIN, FREEMAIN, and MODESET. To improve short term storage acquisition and release storage acquisition was revised to utilize internal cell pool structures.
- Implemented use of z/OS high performance Pause and Release services that replaced WAIT and POST for scheduling the I/O tasks, and replaced SUSPEND and RESUME used to ready completed I/O requests for batch jobs.
- Revised I/O task dispatching and management, as explained below.

The IAM/RLS I/O task management has been revised to provide improved throughput by using fewer I/O tasks. Rather than attaching a new task when an I/O request is received and there are no available I/O tasks, the I/O request is put on a single common work queue. Then the currently active I/O tasks will pull work from the work queue whenever they are done with the I/O request in progress, eliminating any z/OS services from being involved. I/O requests are added to and removed from the work queue with the PLO instruction, avoiding use of z/OS locks. When an active I/O task finds the work queue empty, it uses the Pause service to cause it to wait.
The IAM/RLS workload manager will manage the number of active I/O tasks between the MINIOTASK and MAXIOTASK values to provide good throughput with the least number of I/O tasks. When an I/O task is no longer necessary to handle the workload, it is placed on an idle queue, where it will remain ready to be reactivated when necessary. When the number of I/O requests on the work queue exceeds internal thresholds, an I/O task will be reactivated if available. If none are available, then an additional I/O task will be attached as long as the total number of I/O tasks is less than the MAXIOTASK value. The values are reported on in the periodical RLSINFO reports if you have them enabled.

It is expected that with these changes will reduce the number of I/O tasks being utilized. To help facilitate that, we encourage users to reduce the MINIOTASK value to 4. You can adjust MINIOTASK to determine what value provides the best throughput for your workload, however making it too high can be counter productive with HiperDispatch enabled.

Record lock contention can result in an increase in active I/O tasks. From the IAM/RLS perspective, a wait on a record lock can be a long wait. When an I/O request has to wait for a record lock, IAM checks to see how many other tasks are waiting for locks. If this new request would result in all active tasks waiting for record locks, it will invoke the workload manager to add another task. If the IAM/RLS region is already at MAXTASK IAM/RLS will prevent a complete locked situation by failing I/O requests at that time that need to wait on a lock with a logical error code of x'18'. While record lock contention is generally low, this could be a factor in how high a value to use for MAXIOTASK.

These changes have resulted in reduction in both TCB and SRB times for the IAM/RLS address space and batch address spaces utilizing IAM/RLS services. The savings can result in decreased elapsed times for jobs utilizing IAM/RLS.
20.02  Multiple IAM/RLS Address Spaces

IAM will allow customers to run multiple IAM/RLS address spaces per MVS image (LPAR or System). There could be several reasons to run multiple IAM/RLS address spaces. For example, one might want to isolate different applications such that a failure with one application grouping will not impact other unrelated applications. Another reason might be for billing purposes, to be able to isolate as much as possible the processing for different applications or customers by eliminating common server address spaces. Or, it could be necessary or desirable for security purposes to have certain applications isolated from others.

Whatever the reason customers that desire to utilize multiple IAM/RLS address spaces will need to have well defined groups of datasets and processing that can be isolated from each other to make this possible. The use of multiple IAM/RLS address spaces is subject to the following restrictions:

1. A batch job, CICS region, or TSO user can only be connected to only ONE IAM/RLS address space. Therefore, all IAM datasets that require IAM/RLS processing for that user address space MUST be handled under the same IAM/RLS address space. This is necessary to avoid problems with coordination of journaling, recovery, syncpoint processing and the associated overhead involved in such if multiple IAM/RLS address spaces are involved.

2. An IAM dataset can only be opened by ONE IAM/RLS address space at a time.

3. Each IAM/RLS address space must be started with its own unique proc and name.

4. Each IAM/RLS address space must have its own set of journals, parameter input, and record lock journal dataset if needed.

5. Each IAM/RLS address space requires its own system LX. A system LX is used to establish the cross memory environment required to use an IAM/RLS address space. Due to system limitations on the number of system LX’s, this will restrict the maximum number of IAM/RLS address spaces that can be used unless the address space is started with the REUSASID=YES parameter on the START command for the IAM/RLS procedure.

6. All IAM/RLS address spaces in a SYSPLEX must use the same DATASET NAME TABLE.
Each IAM/RLS address space will be identified by a unique 4-character identifier within the scope of a SYSPLEX, referred to as the RLSID. The RLSID can be indicated via the RLSID startup parameter for IAM/RLS. If the RLSID startup parameter is not specified, then the IAM Global Options Table will contain a default value for the RLSID, which will be set as IAM is shipped to RLS1. Customers will be able to revise that value as they desire.

During OPEN processing for an IAM dataset, a “default” IAM/RLS will be referred to as necessary. The “default” associated IAM/RLS address space will be determined using the RLSID and based upon the current status of the association with an IAM/RLS address space. Once a batch job, CICS region, or TSO user has opened an IAM dataset for processing through an IAM/RLS address space that IAM/RLS address space will be the only IAM/RLS address space that can provide services to that job, region, or user. If no connection to an IAM/RLS address space has been established, then the IAM/RLS address space to connect to will be selected based on the following criteria, from highest to lowest priority order:

1. From the RLSID specified via an IAM ACCESS Override for this job step that relates to the dataset being opened. The RLSID could be specified on either an override that explicitly specifies the dataset being opened via DDNAME or DSNAME, or if no such override has been given, then by an Override for DD=&ALLDD.

2. If no ACCESS override is applicable, then the RLSID that was specified via an IAM CREATE Override when the dataset was defined or loaded will be used, if any was so indicated.

3. If no RLSID overrides were specified and if the dataset name table processing is indicated then the desired RLSID can be obtained from the dataset name table.

4. If RLS processing is required for this dataset and there has been no specification of an RLSID then the RLSID from the IAM Global Options Table loaded for this job step will be used.

There is only one dataset name table that is to be used and shared among all of the IAM/RLS and IAM/PLEX address spaces within a SYSPLEX. This is different from the prior releases where each IAM/RLS address space had its own table. This change was made to make it easier to manage the table, particularly when there are multiple IAM/RLS or IAM/PLEX address spaces. The revised table also provides for dataset name masking capabilities as well as allowing specification of the desired RLSID.
The implementation method chosen by many users is the dataset name table. This method offers a single point of control over which datasets are going to be utilizing IAM/RLS and eliminates the need for any JCL changes for jobs utilizing IAM/RLS. The requirements to utilize this method are:

1. Building a card image dataset containing the dataset SELECT list and if necessary a dataset EXCLUDE list along with the desired RLSID to handle each SELECT group.
2. Specification of the table via a DD statement in each of the IAM/RLS JCL procedures. Each IAM/RLS procedure MUST specify the same table.
3. Revising the IAM Global Options to RLS=(TABLE,OPTIONAL) or RLS=(TABLE,REQUIRED).
4. Starting an IAM/RLS (or IAM/PLEX) address space on each LPAR that is using IAM files during IPL, or before any IAM files are processed. If RLS=(TABLE,REQUIRED) is used, then no IAM files will be able to be opened until an IAM/RLS or IAM/PLEX address space has initialized a table in virtual storage. With Version 9.2 and above a special startup procedure can be used that will start the IAM VIF and load the dataset name table, but will not actually start IAMRLS or IAMPLEX.

One specifies the RLSID on the SELECT statement like this:

- SELECT RLSID=RLS1,DSN=(IAM.DATASET.ONE, IAM.DATASET.TWO,IAM.APPL1*)
- SELECT RLSID=RLS2,DSN=(IAM.DATASET.THREE,IAM.DATASET.FOUR)

If the RLS Global Option is set to use the dataset name table, then the first 2 datasets and any that match the beginning of the file name IAM.APPL1 will automatically be opened under RLS1, and the second 2 datasets will be opened under RLS2.

A second recommended method of implementing the use of multiple IAM/RLS address spaces is to have all files that are going to be using IAM/RLS services to be defined with the IAM CREATE Override of RLSID=nnnn indicating the IAM/RLS address space is to process it. The primary advantage of this method of implementation is that it eliminates the overhead during open processing of every IAM file for scanning the table to determine eligibility for IAM/RLS and eliminates the need for the ECSA storage used to hold the table. There also is no need to change the JCL of any of the job(s) using this dataset, other than the job that define or reorganizes it. The specification of the CREATE Override of RLSID will indicate that this dataset is requesting / requiring IAM/RLS services from the specified IAM/RLS address space. The RLSID can be revised or eliminated at any time through the use of the DEFINE RECATALOG process.

An alternative implementation method is to use the IAM ACCESS Overrides. Using this method, the jobs and CICS regions that have the datasets requiring other than the "default" IAM/RLS address space will need to have their JCL changed to include an IAM ACCESS Override, for example:

//IAMOVRID DD *
ACCESS DD=&ALLDD,RLSID=xxxx
/*

If there are overrides for additional specific datasets, then they will have to include the RLSID=xxxx override as well.
20.03 IAM/RLS Implementation

This section provides the outline of the procedure to follow to activate and implement IAM/RLS. Innovation recommends that you start out with a very limited implementation, and then as you become comfortable with IAM/RLS, the use of it can be expanded to more datasets. Many of the parameters that control IAM/RLS can be easily modified while IAM/RLS is active, which will minimize the need to deactivate IAM/RLS to broaden the implementation to more datasets. Within any CICS region or batch job, there can be some IAM datasets using IAM/RLS, while other IAM datasets are not using the IAM/RLS services.

Careful planning and consideration must be given to how to recover data when a software or hardware error occurs. This is particularly true if you’ve never used any type of data sharing software in the past. For example, if a batch job abends, you may not be able just to restore a dataset and restart the job because of the potential loss of data from CICS users concurrently updating the dataset. IAM does provide journaling and recovery capabilities that can be used to perform various types of recoveries. You should determine ahead of time what your needs are, which can be different for your different datasets and applications. Careful planning and testing of recovery procedures will be of great benefit to successfully implement IAM/RLS.

The various tasks necessary to implement IAM/RLS are shown below. Please be sure to review all of the sections to make sure that nothing is missed as you develop your implementation plan.

- Set up the IAM/RLS proc within the appropriate system proclibs, and allocate any required parameter libraries. Refer to Section 20.05 for the details on the IAM/RLS proc and address space requirements. A separate proc will need to be created for each IAM/RLS address space that is to be executed.

- Define a Record Lock Recovery dataset for each IAM/RLS address space that will need persistent lock support. Refer to Section 20.06 for the details on performing this setup.

- Review and select the appropriate values for the various IAM/RLS parameters. Refer to Section 20.10 for detailed information on the IAM/RLS parameters.

- Set up the appropriate criteria for your installation for the IAM/RLS automatic dataset selection. Refer to Section 20.20 for the details on automatic dataset selection.

- Allocate the IAM/RLS log datasets required to support the journaling that you need IAM/RLS to perform. Set up the procedures to offload the log datasets, and retain the data for subsequent recoveries as might be needed. Refer to Section 20.30 for information on IAM/RLS Journaling. Unique sets of journals will need to be allocated for each separate IAM/RLS address space that is to be used.

- Review Section 20.50 on CICS Transaction Server considerations. Install and activate the necessary CICS exits provided by IAM in every region that will be using IAM/RLS. The CICS exits are critical for IAM to perform record locking for CICS. Failure to properly install and implement them will result in record lockouts and other failures. There are no application or JCL changes required for CICS to use IAM/RLS.
• Review Section 20.60 on batch application considerations. For batch jobs that update a large portion of the records in any recoverable file that is processed by IAM/RLS, seriously consider using the IAM/RLS Batch Syncpoint program, along with providing for recovery and restart should such programs experience software or hardware failures.

• Plan and develop procedures for data recovery, to be prepared in case of unexpected software or hardware errors occur. Make sure that your implementation for IAM/RLS journaling will provide the necessary data to perform data recoveries that will be necessary when failures occur. Refer to Section 20.70 for information on recovery of data for IAM files being processed under IAM/RLS.

• Review Section 20.40 on the IAM/RLS operator commands, for information on the various operational requests that can be made of IAM/RLS.
Setting up IAM/RLS (Record Level Sharing)

IAM/RLS runs as a started task in an MVS system. It has a number of differences from the IAMSTART procedure, which just installed some IAM modules into Dynamic LPA, and terminated. The IAM/RLS procedure runs continuously as a non-swappable address space, providing services to the various jobs and CICS regions that are sharing IAM files. In general, IAM/RLS will be a task that should be kept running all the time. However, IAM/RLS can be shut down and restarted whenever it may be necessary to do so without impacting processing for IAM datasets that are not using IAM/RLS.

Because IAM/RLS is an I/O service provider, you will need to **make sure that the IAM/RLS address space is provided with a high enough priority to service the CICS transactions and batch jobs to meet their response time requirements.** It probably should be given a priority equal to or higher than the most critical CICS region or batch job to insure that adequate service is provided. Specifying too low of a priority will result in poor response times for CICS transactions and elongated run times for batch jobs. If you are using the Workload Manager (WLM) to manage the workload in your SYSPLEX then you should consider assigning the same service class to IAM/RLS that is used for the DB2 Lock Manager.

If you are installing IAM/RLS, you can replace the execution of the IAMSTART procedure with the start up of IAM/RLS. IAM/RLS will automatically run IAMSTART to install the IAM VSAM Interface, if it is not already active. Note that subsequently shutting down IAM/RLS will not deactivate the IAM VSAM interface, the IAM VSAM interface will be left active.

Each IAM/RLS address space will be identified by a unique 4-character identifier, referred to as the RLSID. The RLSID can be indicated via the RLSID startup parameter for IAM/RLS. If the RLSID startup parameter is not specified, then the IAM Global Options Table will contain a default value for the RLSID, which will be set as IAM is shipped to RLS1. Customers can revise that value as they desire.
IAM/RLS Proc

In the ICL (Installation Control Library) that was copied from the product tape, is an example of the IAM/RLS procedure, which is shown below. This example can be edited and copied into one of your system proclibs. You will need to change the data set name of the IAM load library, and the data set name of the card image PDS to be used for parameters and overrides. You may also desire to change the default member names for the IAM/RLS parameters and for the IAM/RLS override.

```
//IAMRLS  PROC MBR=RLSPARM1,OVR=RLSOVRID,TBL=DSNTB
//IEFPROC  EXEC PGM=IAMRLS,
  //      REGION=0M,DPRTY=(15,15),TIME=1440
//STEPLIB  DD   DSN=your.iam.loadlib,DISP=SHR
//SYSLIB   DD   DSN=your.iam.loadlib,DISP=SHR
//SYSPRINT DD   SYSOUT=X
//RLSLOGDD DD   SYSOUT=X
//SYSABEND DD   SYSOUT=X
//RLSINFO  DD   SYSOUT=X
//IAMINFO  DD   SYSOUT=X
//RLOKJRNL DD   DISP=SHR,DSN=your.lock.recovery.dataset
//IPARMLIB DD   DISP=SHR,DSN=your.iam.icl(&MBR)
//IAMOVRID DD   DISP=SHR,DSN=your.iam.icl(&OVR)
//IAMDSNTB DD   DISP=SHR,DSN=your.iam.icl(&TBL)
```

Figure 20-1: Sample of IAM/RLS PROC: Member IAMRLS in ICL
Should a need arise to test PTF's that you have received, or to test an updated
maintenance level of IAM, you can start a test version of the IAM/RLS address space.
This will allow you to limit use of that IAM/RLS address space to specified jobs. Starting
an IAM/RLS for test will also cause a test version of the IAM VSAM Interface to be
started. To start a test version, add a parameter value to the execute statement of
started task JCL, or specify it on a start command. This works the same as the test
versions of IAM worked in the past. A job name or a mask with a job name prefix can
be used which will allow the specified job names to be processed by the IAM/RLS
address space. Up to four (4) different job name(s) can be specified. The test IAM/RLS
address space must have a different name than the production IAM/RLS address
space, which will mean setting up a different proc for the test version. Note that an IAM
data set that is being processed under the production IAM/RLS address space will not
be able to be opened under the test IAM/RLS address space.

An example of a test IAM/RLS proc is shown below. The test job names are specified by
the JOB parameter when starting the IAM/RLST procedure. The “test” IAM/RLS will
need to have a different parmlib member so that different journal data sets can be
specified.

```
//IAM/RLST PROC MBR=RLSPARMT, OVR=RLSOVRID, TBL=DSNTB
//IEFPROC EXEC PGMP=IAMRLS,
//    PARM='TEST,JOBNAME=JOB*',
//    REGION=0M, DPRINT=(15,15),TIME=1440
//STEPLIB DD DSN=your.iam.loadlib, DISP=SHR
//SYSLIB DD DSN=your.iam.loadlib, DISP=SHR
//SYSPRINT DD SYSOUT=X
//RLSLOGDD DD SYSOUT=X
//SYSABEND DD SYSOUT=X
//RLSINFO DD SYSOUT=X
//IAMINFO DD SYSOUT=X
//RLOKJRNL DD DISP=SHR, DSN=your.lock.recovery.dataset
//IPARMLIB DD DISP=SHR, DSN=your.iam.icl(&MBR)
//IAMOVRID DD DISP=SHR, DSN=your.iam.icl(&OVR)
//IAMDSNTB DD DISP=SHR, DSN=your.iam.icl(&TBL)
```

Figure 20-2: Sample of Test IAM/RLS PROC
### IAM/RLS DD Statements

The DD statements needed in the IAM/RLS proc are described below.

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Specifies the IAM load library data set. This DD is optional if IAM is in the system link list.</td>
</tr>
<tr>
<td>SYSLIB</td>
<td>Required DD statement that specifies the IAM load library data set.</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>Required DD statement, specifying a sequential output data set that is normally printed. If specified as SYSOUT, make sure the class used is one that is not automatically purged, in case the output is needed for problem diagnosis.</td>
</tr>
<tr>
<td>RLSLOGDD</td>
<td>Optional DD statement, specifying a sequential output data set that is normally printed. This file contains a log of activity and error messages. If specified as SYSOUT, make sure that the class used is one that is not automatically purged, in case the output is needed for problem diagnosis. If not included in the IAM/RLS PROC, then IAM/RLS will dynamically allocate a printed log file to SYSOUT, using the SYSOUT class value specified by the SYSOUT=x IAM/RLS parameter. If the SYSOUT class is not specified, then the default SYSOUT class for the started proc will be used. By allowing this data set to be dynamically allocated, you can spin off the current log data at any time by using the IAM/RLS LOGSWITCH command.</td>
</tr>
<tr>
<td>SYSABEND</td>
<td>An optional, but highly recommended DD statement, that will be used when a dump is taken due to an abend condition. An alternative would be to use a SYSMDUMP DD statement. Use of SYSUDUMP is not recommended, because it will contain insufficient information for problem diagnosis.</td>
</tr>
<tr>
<td>RLSINFO</td>
<td>An optional DD statement that is required only if you want the IAM/RLS stats reports to be generated as specified by the STATS parameter in the startup parameters.</td>
</tr>
<tr>
<td>RLOKJRNL</td>
<td>An optional DD statement that is required to utilize the record lock recovery / persistence function. See Section 20.06 for detailed description, allocation and usage guidelines.</td>
</tr>
<tr>
<td>IAMINFO</td>
<td>An optional, but highly recommended DD statement that will contain activity reports for each IAM file that is produced whenever that file is closed. If specified as SYSOUT, make sure the class used is one that is not automatically purged, in case the output is needed for problem diagnosis. An alternative to using the printed output is to enable the IAM SMF record, which will save a machine-readable format of the same data produced on the IAMINFO report. The IAMINFO reports can then be obtained from the systems SMF data using the IAMSMF program.</td>
</tr>
<tr>
<td>IPARMLIB</td>
<td>An optional DD statement, which specifies an 80-character card image input data set containing various parameters and options for IAM/RLS. This data set is typically a PDS.</td>
</tr>
<tr>
<td>IAMOVRID</td>
<td>An optional DD statement that specifies an 80-character card image input data set, which contains the IAM override statements. This provides a mechanism for indicating parameters, such as number enabling journaling for all files processed by IAM/RLS.</td>
</tr>
<tr>
<td>IAMDSNTB</td>
<td>An optional DD statement that specifies an 80-character card image input data set. This data set contains the dataset selection and dataset exclusion lists for IAM/RLS processing.</td>
</tr>
<tr>
<td>ALTSYSLG</td>
<td>Optional DD statement that specifies the name of a System Logger logstream to be used for replication logging.</td>
</tr>
<tr>
<td>ABNLIGNR</td>
<td>A DD DUMMY to bypass Abend-Aid dumps.</td>
</tr>
<tr>
<td>ABNLDUMP</td>
<td>A DD DUMMY to bypass ABEND-Aid dumps.</td>
</tr>
</tbody>
</table>
20.06 IAM/RLS Record Lock Recovery

Record lock recovery is an optional process that enables IAM/RLS to reestablish the record locks for recoverable files when an IAM/RLS address space is restarted after a shut down or failure. This function enhances the data integrity by providing persistent record locking within the IAMPLEX sphere, which will enable recovery of the work that was in process at the time of the incident prior to any other attempts to update those records through IAM/RLS. Using a two phase recovery process, as explained below, IAM will re-establish the record locks as they were at the time of failure based on data from both the record lock journal and the IAM/RLS System Logger journal.

Eligibility requirements for record lock recovery are that the locks are being held are for a recoverable IAM data set, which is defined as being a data set that is using the IAM/RLS journaling that includes before images. When record lock recovery is active, it will also automatically result in before images for CICS recoverable files to be written in the IAM/RLS journals. The journal records are necessary for the lock recovery process.

How Lock Recovery Works

The record lock recovery process retains information about the record locks that are currently held on disk in an IAM dataset that is referred to as the Record Lock Journal. The presence of the Record Lock Journal in the IAM/RLS JCL is the indication that the record lock recovery process is to be activated. Each IAM/RLS address space will need to have it’s own Record Lock Journal data set. To minimize the performance impact, the I/O on the data set utilizes deferred writes which delays the actual physical writing of the record lock status. At periodic intervals, referred to as the record lock checkpoints, all the buffers containing updated record lock data are written out to DASD and the current position in the IAM/RLS journal is recorded.

The checkpoint interval can be set through the IAM/RLS parameter LOCKCHKPT, which will specify the checkpoint interval in minutes. The default if the LOCKCHKPT parameter is not provided is 5 minutes. Using shorter times may increase the overhead during regular processing, but decrease the recovery time. User longer checkpoint times will have the opposite effect.

The actual record lock recovery process is done in two phases while the IAM/RLS address space is starting up. The record lock recovery must complete before the IAM/RLS address space will start processing I/O requests. The first phase is to read through the Record Lock Journal dataset and establish the record locks based on the information from that dataset. If the status indicates that the IAM/RLS address space had been shut down properly, meaning that the lock recovery journal has all of the lock data, then the recovery is complete.

If the IAM/RLS address space was not normally shut down, then the record lock recovery process will go into the second phase. The second phase will read the IAM/RLS journal. Either the sequential journal datasets or the System Logger can be used with record lock recovery. Positioning is established at the place in the journal where the last record lock checkpoint had occurred, then proceed forward in the journal from that point. The recovery process will then update the record locks based on the activity indicated by the journal records, which includes updating the lock recovery journal.
It is recommended that users backup the lock recovery journal prior to restarting the IAM/RLS address space from a failure. This can be done with a copy, a backup with FDRDSF, ABR or similar product, or by an IDCAMS REPRO. If a failure should occur on the restart of the IAM/RLS address space, one will then be able to restore the lock recovery journal and retry the restart process again, otherwise a cold start may be necessary.

On completion of the record lock recovery, a summary of the activity performed by the lock recovery process will be printed on the RLSLOGDD dataset. Then the IAM/RLS address space will finish initialization and start accepting requests. To find out what record locks have been recovered one can either use the IAM/RLS display retained locks command or use the IAMBMON facility under the IAM ISPF panels. If any of the work is not going to be recovered, the associated locks can be released using the IAM/RLS release locks command.
To activate the record lock recovery function, a lock recovery journal dataset must be pre-allocated using an IDCAMS DEFINE prior to use for each IAM/RLS address space that record lock recovery is going to be used. The lock recovery journal dataset contains a control record, lock owner records, and record lock records. The dataset will be formatted when it is first used based on the IAM/RLS parameters of MAXJOBS, MAXLOCKS, and MAXTRANS. Below is an example of the JCL and control cards that are used to define the lock recovery journal dataset:

```java
//DEFRLKJR EXEC PGM=IDCAMS
//IAMPRINT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//IAMOVRID DD *  
CREATE DD=&ALLDD,DATACOMP=NO,DSNTYPE=LARGE,RELEASE=NO  
//SYSIN DD *  
DELETE (IAMV.RLOKJRNL.rlsid.CLUSTER)  
DELETE (IAMV.RLOKJRNL.rlsid.CLUSTER) NOSCRATCH  
SET MAXCC=0  
/* RECOMMENDED SPACE ALLOCATION VALUES: */  
/* CYL(  60       6) WITH MAXLOCKS UP TO  133,103 */  
/* CYL( 120      12) WITH MAXLOCKS UP TO  266,239 */  
/* CYL( 230     23) WITH MAXLOCKS UP TO  532,453 */  
/* CYL( 460     46) WITH MAXLOCKS UP TO 1,064,907 */  
/* CYL( 900     90) WITH MAXLOCKS UP TO 2,129,803 */  
/* CYL(1350    135) WITH MAXLOCKS UP TO 3,193,783 */  
/* CYL(1800    180) WITH MAXLOCKS UP TO 4,194,301 */  
DEFINE CLUSTER -  
(NAME(IAMV.RLOKJRNL.rlsid.CLUSTER) -  
CYL( 900  90) -  
VOL(* * * ) -  
CISZ(12288) REUSE -  
STORCLAS(TSTDATA) -  
OWNER($IAM) -  
RECORDSIZE(100 328) KEYS(5,0) -  
FREESPACE(0,0) -  
SHAREOPTIONS(2,3))  
LISTC ENTRIES(IAMV.RLOKJRNL.rlsid.CLUSTER) ALL  
/*
```

Figure 20-3: Example of JCL for Lock Recovery Journal, Member LKRCVJRNL in ICL

When using the above JCL, users will need to change the dataset name and the STORCLAS. Also the space requested should be updated based on the MAXLOCKS value they are using. Please note that one needs to use the next larger value from the table if the MAXLOCKS value chosen exceeds the prior value. For example if MAXLOCKS=133104 is specified, then the space allocation used must be the one for up to 266,239 lock records. All of the other parameters and overrides shown in the example above must be used to provide proper functioning and performance of the lock recovery function. If the value for MAXLOCKS is ever increased, the lock recovery journal should be re-allocated with the required space values indicated.
After defining the lock recovery journal, users must then make the following changes to the IAM/RLS JCL:

- Add the RLOKJRNL DD statement to the IAM/RLS JCL specifying the newly defined dataset.
- Add a card in the IAMOVRID member input that specifies the following:
  
  ACCESS DD=RLOKJRNL,NORLS,JRNAD=NONE

When the IAM/RLS address space is started, IAM/RLS will automatically determine that the dataset is unloaded and format the dataset with the appropriate number of pre-formatted records. This pre-formatting is completed quickly and will provide performance benefits for the operation of the lock recovery function. If any of the key IAM/RLS parameters are changed, that is MAXLOCKS, MAXJOBS, or MAXTRANS, then the dataset should be re-initialized by specifying the IAM/RLS parameter of LOCKSTART=COLD which will trigger the reformatting of the dataset. If the value for MAXLOCKS has been increased then the dataset may need to be deleted and re-defined with the appropriate amount of space.

It is highly recommended that a backup copy of the lock recovery journal be taken whenever the IAM/RLS has come down, whether it was scheduled or unscheduled. The dataset can be backed up with a copy, FDRDSF, ABR or similar type of backup, or by an IDCAMS REPRO. Then if there is a problem on the restart, the lock recovery journal can be restored prior to retrying the restart.

Backups can be taken at other times while the IAM/RLS address space is running, however the restored copies can not really be used to restart the IAM/RLS for other than a cold start (LOCKSTART=COLD). This is because the restored copy is far from being synchronized with the actual state when the IAM/RLS address space had terminated. A cold start is also likely to be required when starting IAM/RLS address spaces at disaster recovery sites

There generally should not be any need to reorganize the lock recovery dataset. The define parameters and overrides in the example above were carefully chosen along with the way the dataset is processed to minimize the need for use of the overflow area. This is particularly true when appropriate values were specified on the IAM/RLS parameters such that they were sufficient to meet the processing that was done. One can either look at a LISTCAT output or at the IAMINFO report to determine if there are a lot of records in overflow. If there are, then one can check on the maximum locks used from the RLSINFO report that is produced or from the IAMBMON display to make sure that sufficient lock records had been formatted. One should also check that all of the appropriate parameters in the define example above were used. It may be necessary to delete and redefine the dataset such that the overflow area usage will be minimized. Innovation technical support can provide assistance if needed.
20.10 IAM/RLS Parameters

There are three sources of parameters for controlling IAM/RLS usage. These include the IAM/RLS start up parameters, specified by a parmlib mechanism on the IPARMLIB DD statement, the IAM Overrides, and the IAM Global Options Table. The IAM/RLS parmlib parameters are completely discussed in this section. There is also a discussion of relevant IAM Overrides with IAM/RLS in this section, the full IAM Override documentation is provided in Section 30 of this manual. The main global option affecting IAM/RLS has to do with specification of automatic eligibility for IAM/RLS processing, which is discussed in Section 20.20.

IAM/RLS PARMLIB

The parameters that can be specified for the IAM/RLS address space are presented below. These parameters are passed to IAM/RLS via a parmlib mechanism, which is specified via the IPARMLIB DD statement. If you use the example IAM/RLS procedure, then the IAM/RLS parmlib member is specified using the MBR= operand. You will need a different parmlib member for each unique system on which IAM/RLS is being run to specify different log data sets.

The parmlib is read to establish the parameter settings during IAM/RLS initialization. Additionally, customers can issue the “CHANGEPARM” command, which will cause an active IAM/RLS address space to reread the parmlib member, and cause an immediate update for many of the values. Any special considerations with changing the various parameter values will be discussed in the description of the individual parameters below.

The parmlib dataset must be an 80-column card image dataset, with parameters and values coded in columns 1 through 71. Each card image can contain a single parameter with its values. Multiple parameters can be specified on a single card, but must be separated by a comma and no spaces.

Comment cards can be included, and are indicated by an “*” in the first column of the card. Examples of parmlib are provided in the ICL (Installation Control Library) that was copied from the installation tape. Customers may use the ICL for the parmlib data set, if so desired.

IAM/RLS Parameters

Shown below are all of the IAM/RLS parameters that can be specified in the IAM/RLS PARMLIB, along with the description of each parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICSJOURNAL=</td>
<td>Specifies whether IAM should journal the before images for updates performed by CICS transactions. Valid values are YES or NO. The default value is NO, because CICS normally handles transaction backout and before image journaling itself. If the Record Lock Recovery is being used, then it will automatically default to YES.</td>
</tr>
</tbody>
</table>
**CICSLOCK=**

Specifies the action that IAM/RLS will take when record lock contention is encountered for a CICS transaction. Three sub-parameters can be provided. The first instructs IAM/RLS what action to take when the owner of the lock is another CICS transaction. The second instructs IAM/RLS what action to take when the owner of the lock is other than a CICS transaction, such as a batch job or TSO user. The third tells IAM/RLS that if it is going to fail the request due to lock contention, it is just to pass a logical error code back to the caller, or abend the transaction.

One of the following keywords can be provided to indicate what action IAM/RLS will take when the owner of the record lock is a CICS transaction:

- **CICSWAIT** specifies that the requester of the lock will wait until the record lock is available when the owner is a CICS transaction.

- **CICSTIMEOUT** specifies that the requester of the lock will wait for a short time as specified by the CICSTIMEOUT parameter (see below), if the owner of the lock is a CICS transaction. If the lock is still not available, then the request will fail.

- **CICSNOWAIT** specifies that the request for the record lock will be failed if the owner is a CICS transaction.

One of the following keywords can be provided to indicate what action IAM/RLS will take when the owner of the record lock is other than a CICS transaction, such as a batch job or TSO user:

- **WAIT** specifies that the requester of the lock will wait until the record lock is available when the owner is not a CICS transaction.

- **TIMEOUT** specifies that the requester of the lock will wait for a short time as specified by the CICSTIMEOUT parameter (see below). If the lock is still not available, then the request will fail.

- **NOWAIT** specifies that the request for the record lock will be immediately failed when the owner is not a CICS transaction.

One of the following keywords can be provided to indicate whether IAM should fail the CICS transaction with an ABEND, or just fail the request following standard VSAM protocol with a logical error code:

- **ABEND** specifies that IAM will abend the CICS transaction. This requires the installation and activation of an IAM provided CICS Global User Exit, as described later. The transaction will abend with a code of AKC3.

- **NOABEND** specifies that IAM will fail the request with a logical error code.

The default values are: CICSLOCK=(CICSWAIT,TIMEOUT,NOABEND)

**CICSTIMEOUT=**

Specifies the maximum time, in seconds that a CICS transaction will wait for a record lock if CICSTIMEOUT or TIMEOUT has been specified for the CICSLOCK parameter. Default is 30 seconds.
<table>
<thead>
<tr>
<th><strong>Parameter</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>DJB=</td>
<td>This parameter provides a way to control Dynamic Job Backout. The following values are accepted:</td>
</tr>
<tr>
<td></td>
<td><strong>YES</strong> – Specifies that Dynamic Job Backout will be automatically invoked for any job steps that abend, and have recoverable IAM files accessed through IAM/RLS. Changes made to recoverable IAM files for the abending job step will be removed from the affected files, back to either the most recent batch syncpoint taken by the job step, or if no syncpoints were taken, then to the beginning of the job step.</td>
</tr>
<tr>
<td></td>
<td><strong>NO</strong> – Specifies that Dynamic Job Backout will not be automatically invoked, unless the failing job step has opened a recoverable IAM file through IAM/RLS with the DJB=YES IAM override.</td>
</tr>
<tr>
<td></td>
<td><strong>DISABLED</strong> – Dynamic Job Backout support is disabled, and will not be invoked even with a DJB=YES IAM override.</td>
</tr>
<tr>
<td></td>
<td>The default value is NO.</td>
</tr>
<tr>
<td>JRNBUFF=</td>
<td>Optional parameter that specifies the number of buffers that IAM/RLS will use for journaling. Valid values are from 2 to 255. Low activity systems can use fewer buffers, while more active systems will benefit from additional buffers.</td>
</tr>
<tr>
<td></td>
<td>The default value is 32.</td>
</tr>
<tr>
<td>JRNPROC=</td>
<td>Optional parameter that specifies the name of a procedure (proc) in the system proclibs which IAM/RLS will invoke via a START command whenever a log data set is filled. The name of the log dataset will be passed via the LOG= parameter on the start command. Use of this parameter will aid in automatically backing up and emptying the log datasets in a timely manner, to avoid potential delays should all of the log datasets be filled up.</td>
</tr>
<tr>
<td></td>
<td>The default value is that no procedure will be automatically started by IAM/RLS when a log fills up, only a WTO message will be issued. Refer to Section 20.30 IAM/RLS Journaling and Section 20.70 IAM/RLS Recovery for further information and recommendations regarding this parameter.</td>
</tr>
</tbody>
</table>
LOCK=

Specifies the action that IAM/RLS will take when record lock contention is encountered for a requester other than a CICS transaction, such as a batch job or TSO user. Three sub-parameters can be provided. The first instructs IAM/RLS what action to take when the owner of the lock is a CICS transaction. The second instructs IAM/RLS what action to take when the owner of the lock is other than a CICS transaction, such as a batch job or TSO user. The third tells IAM/RLS that if it is going to fail the request due to lock contention is it to pass a logical error code back to the caller, or abend the job or TSO user.

One of the following keywords can be provided to indicate what action IAM/RLS will take when the owner of the record lock is a CICS transaction:

- **CICSWAIT** specifies that the requester of the lock will wait until the record lock is available when the owner is a CICS transaction.

- **CICSTIMEOUT** specifies that the requester of the lock will wait for a short time as specified by the LOCKTIMEOUT parameter (see below), if the owner of the lock is a CICS transaction. If the lock is still not available, then the request will fail.

- **CICSNOWAIT** specifies that the request for the record lock will be failed if the owner is a CICS transaction.

One of the following keywords can be provided to indicate what action IAM/RLS will take when the owner of the record lock is other than a CICS transaction, such as a batch job or TSO user:

- **WAIT** specifies that the requester of the lock will wait until the record lock is available when the owner is not a CICS transaction.

- **TIMEOUT** specifies that the requester of the lock will wait for a short time as specified by the LOCKTIMEOUT parameter (see below). If the lock is still not available, then the request will fail.

- **NOWAIT** specifies that the request for the record lock will be immediately failed when the owner is not a CICS transaction.

One of the following keywords can be provided to indicate whether IAM should fail the requester with an ABEND, or just fail the request following standard VSAM protocol with a logical error code:

- **ABEND** specifies that IAM will abend the batch job or TSO user. The abend code will be a U0185.

- **NOABEND** specifies that IAM will fail the request with a logical error code.

The default values are: LOCK=(CICSWAIT,TIMEOUT,NOABEND)

**LOCKCHKPT=**

Specifies the number of minutes between checkpoints on the lock recovery dataset. Valid values are 1 to 60. Refer to Section 20.06 for information on this parameter.

The default value is 5 minutes.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCKMEM=</td>
<td>64BIT specifies that 64 bit storage will be used for the record lock table. If used a MEMLIMIT must be specified on the EXEC statement in the IAM/RLS proc at startup. If MEMLIMIT is not specified then IAM/RLS will default to using 31 bit storage for the record lock table. 31BIT specifies that 31 bit storage will be used for the record lock table and is the default if no LOCKMEM parameter is provided or if a MEMLIMIT is not specified.</td>
</tr>
<tr>
<td>LOCKSTART=</td>
<td>Specifies the type of start up to be done for the Lock Recovery Process. Valid values are: COLD will result in no attempt to restore the record locks and force the reformatting of the lock recovery dataset. WARM will result in restoring the record lock environment for recoverable files to what it had been at the time when the IAM/RLS region became unavailable. The default value is WARM.</td>
</tr>
<tr>
<td>LOCKTIMEOUT=</td>
<td>Specifies the time, in seconds, that a batch job or TSO user will wait for a record lock to become available if the CICSTIMEOUT and / or TIMEOUT value have been specified for the LOCK= parameter. The default value is 30.</td>
</tr>
<tr>
<td>LOGDSN1 =</td>
<td>Specifies the data set name(s) of up to six log data sets. This is required if journaling is going to be used for any IAM data set under IAM/RLS control, and you are not using the System Logger. These data sets are sequential DASD data sets. IAM/RLS will determine at startup time which journal was last used, and resume journaling with that journal data set. It will then subsequently rotate through the list of journal data sets as each journal is filled. If the next journal data set has not been emptied, or flagged as reusable by IAMJUTIL, IAM/RLS will issue a WTO for the operator to reply to reuse the journal dataset, or to quiesce IAM/RLS processing. Additional information on journaling under IAM/RLS is in Section 20.30.</td>
</tr>
<tr>
<td>LOGDSN2=</td>
<td></td>
</tr>
<tr>
<td>LOGDSN3=</td>
<td></td>
</tr>
<tr>
<td>LOGDSN4=</td>
<td></td>
</tr>
<tr>
<td>LOGDSN5=</td>
<td></td>
</tr>
<tr>
<td>LOGDSN6=</td>
<td></td>
</tr>
<tr>
<td>LOGSTRM=</td>
<td>Specifies the name of the IAM log stream to be used for journaling, when the System Logger is going to be used. This value must match the log stream name defined by IXCMAIPU. This value is only used if the SYSLOGGER=YES parameter has been specified. The default value, if not specified, is IAMLOGR.sysname, where sysname is the value provided at IPL time in the IEASYSxx member for SYSNAME. Using the system logger is not recommended unless your installation has experience with using the system logger for other purposes, and is familiar with how to use it.</td>
</tr>
<tr>
<td>MAXBUFNO=</td>
<td>Specifies a value to be used as the default MAXBUFNO for IAM files opened under IAM/RLS. If not specified, the default will be the normal IAM default from the IAM Global Options Table. Valid values are from 1 to 2048. This value may be overridden for specific files by an IAM ACCESS override cards that have been included with IAM/RLS. If this value is changed when the parameters are reread by the CHANGEPARAM command, it will take effect for datasets opened subsequent to the change. Currently opened datasets will not have their MAXBUFNO value changed.</td>
</tr>
<tr>
<td>MAXIOTASK=</td>
<td>Specifies the maximum number of file I/O tasks to be permitted. This represents the maximum number of concurrent I/O requests that the IAM/RLS address space will be able to be actively processing at any point in time. Additional I/O requests will be delayed pending availability of an I/O task to process it. Values from 8 to 255 can be specified. Default value is 64.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MAXJOBS=</td>
<td>Specifies the maximum number of jobs that can have IAM files opened under the IAM/RLS address space. This is primarily used for setting up the record locking tables. Specifying too small of a value may cause increased overhead for the IAM lock manager if the actual number of jobs exceeds this limit. Values from 1 to 9,999,999 can be specified. Default value is 1024. Changing this value will only take effect when IAM/RLS is restarted.</td>
</tr>
<tr>
<td>MAXLOCKS=</td>
<td>Specifies the approximate maximum number of record locks that are anticipated to be needed at any particular point in time. This value will be used to determine the size of the lock lookup hash table. Batch jobs will not be able to exceed holding this number of locks concurrently. Specifying too small of a value may cause increased overhead for the IAM lock manager. Values from 1 to 67108859 can be specified. When using LOCKMEM=31BIT the default value is 133103 and should not be made larger than 2129903. When LOCKMEM=64BIT the default is 2129903. Changing this value will only take effect when IAM/RLS is restarted.</td>
</tr>
<tr>
<td>MAXTRANS=</td>
<td>Specifies the approximate maximum number of concurrently active transactions that a CICS system can have. Too small a value may cause increased overhead for the IAM lock manager. Values from 1 to 9999999 can be specified. Default value is 512. Changing this value through the CHANGEPARM command will take effect for CICS regions that are started subsequent to the change.</td>
</tr>
<tr>
<td>MESSAGES=</td>
<td>Specifies what messages should be printed to the IAM/RLS RLSLOGDD file. Valid values are ALL, INFO, or ERR. ALL is primarily intended for problem diagnosis, as various messages will be written indicating many frequent events. INFO will provide a detailed activity log, including when data sets are opened and closed. ERR will limit the RLSLOGDD messages to only those issued in error situations. This value can be dynamically changed via operator command. Default value is INFO.</td>
</tr>
<tr>
<td>MINBUFNO=</td>
<td>Specifies a default minimum number of buffers per IAM file. Valid values are from 1 to 2048. The default value is 1, unless otherwise overridden by an IAMOVRID specification. If this value is changed when the parameters are reread by the CHANGEPARM command, it will take effect for datasets opened subsequent to the change. Currently opened datasets will not have their MINBUFNO value changed.</td>
</tr>
<tr>
<td>MINIOTASK=</td>
<td>Specifies the minimum, and also starting, number of tasks in the IAM/RLS address space that will be used to process I/O requests. Values from 1 to 255 can be specified. Default value is 4, which is recommended due to the changes for HiperDispatch.</td>
</tr>
<tr>
<td>RLSID=</td>
<td>Specifies the four character identifier for this IAM/RLS address space. Each IAM/RLS address space on an LPAR / MVS system image will need to have it's own unique identifier. If not specified, the value from the IAM Global Options Table will be used.</td>
</tr>
<tr>
<td>STATS=</td>
<td>When specified as YES, the RLSINFO reports will be generated automatically every 30 minutes if any stats have changed during that time period. The default is NO. The report can always be generated manually via operator command. The RLSINFO DD statement must also be present in the IAM/RLS PROC used to start the IAM/RLS address space.</td>
</tr>
<tr>
<td>SYSLOGGER=</td>
<td>Specifies either YES or NO, to indicate whether the z/OS System Logger is to be used instead of the sequential data sets. Refer to the Section 20.30 on journaling for more information on using this capability. If YES is specified, you must also specify the LOGSTRM= parameter, indicating the name of the log stream. When specified as YES, all of the LOGDSNx parameters are ignored. Default value is NO.</td>
</tr>
</tbody>
</table>
Example of IAM PARMLIB Parameters

An example of parmlib input is shown below:

```plaintext
MINBUFNO=20,MAXBUFNO=100
LOGDSN1=MY.LOGDSN1.JOURNAL
LOGDSN2=MY.LOGDSN2.JOURNAL
LOGDSN3=MY.LOGDSN3.JOURNAL
LOGDSN4=MY.LOGDSN4.JOURNAL
LOGDSN5=MY.LOGDSN5.JOURNAL
LOGDSN6=MY.LOGDSN6.JOURNAL
SYSLOGGER=NO
SYSOUT=H Put RLSLOGDD in held SYSOUT class
MESSAGES=INFO
CICSLOCK=(CICSWAIT,TIMEOUT,NOABEND)
LOCK=(CICSWAIT,WAIT,ABEND)
CICSTIMEOUT=10
LOCKTIMEOUT=1800
LOCKMEM=64BIT Use 64 bit storage for the indexes
LOCKCHKPT=5 Record lock recovery checkpoint 5 min
LOCKSTART=WARM Perform lock recovery on startup
```

Figure 20-4: Example of IAM/RLS Parmlib Parameters (EX2010A)
Abending CICS Transactions

If ABEND was specified for the CICSLOCK parameter, then to cause the abends to occur the IAM provided CICS Global User Exit (GLUE) must be activated. This IAM provided Global User Exit (GLUE) will purge transactions that receive an indication from IAM/RLS to do so. This indication will only be presented when the action specified for the IAM/RLS CICSLOCK parameter is set to ABEND. The transaction will abend with a code of AKC3. For information on setting this up, refer to Section 20.50, CICS Considerations.

IAM Overrides

IAM ACCESS Overrides can be specified for the IAM/RLS address space. The overrides can be a member of a card image PDS, such as the IAM ICL. Some of the IAM ACCESS overrides that may be useful to IAM/RLS are BUFSPACE, JRNAD, MINBUFNO and MAXBUFNO. The IAM CREATE overrides are not relevant to IAM/RLS because no file creations will take place under the IAM/RLS address space.

The BUFSPACE, MINBUFNO or MAXBUFNO can be used on a dataset by dataset basis, to either use more or less buffering than what is being defaulted to under IAM/RLS. The need to alter the buffering values should be minimal. This is because IAM/RLS offers a MAXBUFNO value to be specified as a default for all files opened under IAM/RLS as a start up parameter that was previously described.

The JRNAD override can be used to enable journaling for all IAM files accessed under IAM/RLS, by providing an ACCESS DD=ALLDD,JRNAD=BOTH. Be sure to specify JRNAD on any additional ACCESS override cards provided for specific data sets, as they will not pick up the value from the &ALLDD override. Using the override may be the preferred way to control journaling for IAM/RLS, rather than on an individual file basis. The journaling can be turned off for specific datasets by an additional ACCESS override indicating the dataset name.

If you are using the IAM overrides with IAM/RLS, then IAM will automatically reread the override input file each time an IAM data set is opened under this IAMRLS address space. This will allow you to change the overrides without having to stop and restart IAM/RLS for the new overrides to take effect.

```
ACCESS DD=&ALLDD,INDEXSPACE=64BIT,
ACCESS DD=RLOKJRNL,INDEXSPACE=64BIT,NORLS,JRNAD=NONE
```

Figure 20-5: Example of Overrides for IAMRLS Address Space, Member RLSOVRID in ICL
IAM/RLS Automatic Dataset Eligibility

IAM can automatically decide during OPEN processing which IAM data sets are to be handled by IAM/RLS. Automatic selection is based on the share options that a data set was defined with, if an RLSID was specified when a dataset is defined or loaded, and/or by the data set name table. An installation specifies the criteria IAM is to use through the IAM Global Options Table, via the RLS option. Using the RLS option, users can specify the minimum share option required for eligibility for IAM/RLS processing, and whether or not a data set name table has been provided to IAM/RLS. If a combination of share option and data set name eligibility is requested, the user can specify via the RLS option whether a data set must meet both the share option criteria and be in the data set name table, or if a match on either one sets the data set as eligible for IAM/RLS. The values that can be specified for the RLS Global Option are shown below:

To minimize OPEN overhead in determining a data set’s eligibility for IAM/RLS processing, it is recommended to define files that are to be handled by IAM/RLS with the IAM CREATE Override of RLSID. The use of the dataset name table processing will add overhead to every IAM dataset that is opened for the searching of the dataset name exclude and include lists.

**RLS Global Option**

Multiple values can be specified for the RLS Global Option Parameter. This parameter indicates how IAM/RLS and IAM/PLEX will be selecting the IAM datasets to place under their control.

<table>
<thead>
<tr>
<th>RLS Option Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>Indicate that to be automatically eligible for IAM/RLS, a data set must have the specified share option <strong>AND</strong> must be selected in the dataset name table.</td>
</tr>
<tr>
<td>NONE</td>
<td>Indicates that no data sets are to be considered for automatic eligibility for IAM/RLS. Set this option if you are not going to be activating the IAM/RLS address space, or if you want to manually direct activity to IAM/RLS through the IAM Overrides. This is the default option as shipped.</td>
</tr>
<tr>
<td>OPTIONAL</td>
<td>Allows data sets to be opened without IAM/RLS when the IAM/RLS address space is not active. This is useful for when the TABLE option has been specified and IAM/RLS may not be active on all LPAR’s because IAM/RLS has to at least have been active at some point in time for the data set include and exclude tables to be loaded. If specified, this value will also be applied to datasets defined with an RLSID, in that if IAMRLS is not active, the dataset will be opened locally.</td>
</tr>
<tr>
<td>OR</td>
<td>Indicates that to be automatically eligible for IAM/RLS, a data set must have either the specified share option <strong>OR</strong> be in the data set name select list. If a data set meets the share option eligibility, then the dataset name table will not be examined. If a data set does not meet the share option eligibility criteria, then it must selected in the dataset name table to be selected for IAM/RLS processing. OR is the default if both TABLE and a SHAREx value are specified.</td>
</tr>
<tr>
<td>REQUIRED</td>
<td>Indicates that IAM/RLS MUST BE ACTIVE if a data set is eligible for IAM/RLS processing, or when the TABLE Global Option is specified. If IAMRLS is not active, then any attempt to open a potentially IAM/RLS eligible data set will be failed. REQUIRED is the default. For datasets defined with an RLSID, IAM/RLS must be active even if RLS=NONE is set, unless RLS=(NONE,OPTIONAL) is specified.</td>
</tr>
<tr>
<td>SHARE1</td>
<td>Indicates that data sets with any cross-region share option value (1, 2, 3, or 4) will be eligible for IAM/RLS processing. Use of this value is not recommended.</td>
</tr>
</tbody>
</table>
Using the RLS Global Option

The following discussion applies to the use of the RLS Global Option. Full use of the IAM Global Option Change Facility is described in Section 91 of the IAM Users Manual. Multiple values can be specified for the RLS option, by enclosing them within parenthesis. When the RLS option is specified on the ZAP control statement, the value(s) specified will act as a complete replacement for the existing RLS option. As shipped, the default value is RLS=NONE, meaning that no IAM files are automatically eligible for IAM/RLS processing.

To use only the dataset name table for automatic dataset selection specify the RLS Global Option of either RLS=(TABLE,OPTIONAL) or RLS=(TABLE,REQUIRED). Use of the REQUIRED option will make sure that files selected for RLS will not open if RLS is not available. Also note with required that no IAM files can be opened unless an IAM/RLS address space has been started since the last IPL and initialized the dataset selection table in ECSA. Once the table is initialized, the IAM/RLS and IAM/PLEX address spaces can be started and shutdown as desired, the table will still remain in storage.

You might desire some combination of selection criteria; for example, you want all share option 3 or 4 files to be eligible, and some selected share option 2 files. For this, you would specify RLS=(SHARE3,OR,TABLE). You would then specify the name(s) of the other files to be included or excluded in the data set name tables. The reason for both an include and an exclude list is so that you could select all data sets beginning with PROD.SHARE for IAM/RLS processing. You would specify in the data set name table SELECT DSN=PROD.SHARE** (ending with a double *). However, perhaps there is a file with that name prefix that you want excluded from IAM/RLS, so you could specify EXCLUDE DSN=PROD.SHARE.FILE1.

Another possibility is that you might want only share option 3 or 4 files to be eligible for IAM/RLS, however you might want to exclude some selected files. In this situation, you would specify RLS=(SHARE3,AND,TABLE). Then in the data set name table, you would specify SELECT DSN=* which will include all data sets, and then exclude specific data sets, such as EXCLUDE DSN=PROD.DONOT.SHARE.FILE1.

<table>
<thead>
<tr>
<th>RLS Option Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHARE2</td>
<td>Indicates that data sets with cross-region share options of 2, 3, or 4 will be eligible for IAM/RLS processing. Use of this value is not recommended.</td>
</tr>
<tr>
<td>SHARE3</td>
<td>Indicates that data sets with cross-region share options of 3 or 4 will be eligible for IAM/RLS processing.</td>
</tr>
<tr>
<td>SHARE4</td>
<td>Indicates that only data sets with a cross-region share option of 4 will be eligible for IAM/RLS processing.</td>
</tr>
<tr>
<td>TABLE</td>
<td>Indicates that IAM is to search the dataset name table to determine eligibility for IAM/RLS processing, subject to other criteria. Be careful of using the options TABLE with REQUIRED. With both of those options, no IAM data sets can be opened until an table has been loaded into virtual storage. If IAMRLS is not started up immediately during the IPL process, consider starting the IAMRLS with PARM=DSNTLOAD which will start VIF and load the DSN TABLE then exit. The default is that the data set name tables will not take part in eligibility selection for IAM/RLS.</td>
</tr>
</tbody>
</table>

SHARE2 Indicates that data sets with cross-region share options of 2, 3, or 4 will be eligible for IAM/RLS processing. Use of this value is not recommended.
SHARE3 Indicates that data sets with cross-region share options of 3 or 4 will be eligible for IAM/RLS processing.
SHARE4 Indicates that only data sets with a cross-region share option of 4 will be eligible for IAM/RLS processing.
TABLE Indicates that IAM is to search the dataset name table to determine eligibility for IAM/RLS processing, subject to other criteria. Be careful of using the options TABLE with REQUIRED. With both of those options, no IAM data sets can be opened until an table has been loaded into virtual storage. If IAMRLS is not started up immediately during the IPL process, consider starting the IAMRLS with PARM=DSNTLOAD which will start VIF and load the DSN TABLE then exit. The default is that the data set name tables will not take part in eligibility selection for IAM/RLS.
The data set name table is provided to the IAM/RLS address space through the IAMDSNTB DD statement. This DD must specify a card image data set, either sequential or a PDS member. The table is read in during initialization of the first IAM/RLS or IAM/PLEX address space started on an LPAR, and stored in ECSA. The table is kept in common to minimize the overhead of searching the table during IAM OPEN processing. Customers can update the data set name tables within the IAMDSNTB data set, and activate the updated tables by using the CHANGEDSNT command of the IAM/RLS address space. If you have IAM/RLS running on multiple LPARs, you will have to issue the command on one IAM/RLS address space on each LPAR.

Within the IAMDSNTB data set, two lists of data set names can be provided. Each data set name entry is considered a full data set name unless it contains 1 or more of the special masking characters. The masking characters allow a single entry in the table to represent multiple dataset names, resulting in a smaller table. Refer to the topic below on dataset name masking for complete information. The old use of a trailing single asterisk (*) will still work the same as it did previously.

The first list is the data set name selection list. It is specified by "SELECT RLSID=rlsx,DSN= (" followed by a list of data set names, separated by commas. All of the datasets included on that select statement are assigned to the specified RLSID. Multiple cards can be used, however each data set name or mask must be fully contained within a single card image. A closing parenthesis is used at the end of the last data set name. If there is just one data set name, or only the * (single asterisk) then no parenthesis are needed.

The exclude list is specified by the "EXCLUDE DSN= (" followed by a list of data set names, separated by commas. Multiple cards can be used however each data set name or mask must be fully contained within a single card image. A closing parenthesis is used at the end of the last data set name. If there is just one data set name, or only an **, then no parenthesis are needed. The EXCLUDE statement must start on a new card.

The EXCLUDE list is always searched first, so if the dataset name matches any mask in the exclude list it is automatically not eligible for IAM/RLS processing even if it is in the INCLUDE list. As much as possible, it is recommended that the EXCLUDE list be kept short but very broad so that as many datasets as possible will be recognized as not requiring IAM/RLS or IAM/PLEX processing.
You can group dataset names by RLSID. If no RLSID is specified on the SELECT statement then it will be blank in the table and during open processing the RLSID provided by the CREATE or ACCESS overrides or the default in the Global Options Table will be used. There can be multiple SELECT DSN= and EXCLUDE DSN= statements in the IAMDSNTB DD input file. Each statement will be processed with the dataset names added to the corresponding Select or Exclude table in storage. If an error is encountered on the first SELECT or EXCLUDE statement, and there was already a Select or Exclude table in storage, then the old table will still be in effect. If an error is encountered on a subsequent SELECT or EXCLUDE statement in the processing of the IAMDSNTB DD input file then the in storage Select and Exclude table will contain only those dataset names that were successfully processed on the previous input cards before the error was encountered. A maximum of 399 input cards can be specified for each SELECT or EXCLUDE statement with multiple dataset names on each card. A total maximum of 32,767 names can be specified for each list. However, it is highly recommended that you keep the lists as short as possible, to keep the ECSA storage requirement low, and to minimize the searching CPU time. For additional information on acceptable formats, refer to Section 1.04 of the IAM Users Manual for general information on control statement format.

To provide more flexibility and help reduce the size of the table, IAM provides a masking facility that can be used when specifying dataset names on the SELECT and EXCLUDE statements. The mask may contain:

- Any valid alphanumeric or national character representing itself
- / (slash) symbol represents a single valid character
- | (vertical bar) represents a single alphabetic character
- + (plus) represents a single valid numeric character
- ? (question mark) represents a single valid national character i.e. (#, $, or @ in the US)
- * (single asterisk) represents zero or more valid characters within one index level except when it is the last character in the mask in which case it is treated as a meaning any subsequent characters.
- ** (double asterisk) represents zero or more valid characters contained in one or more index levels (including their periods)
- . (period) represents a period (index level) in the dataset name except as noted in special cases below. A period at the beginning of the mask, or immediately following another period is interpreted as an *. (single asterisk, period).
- **. (double asterisk, period) at the beginning of a string represent one or more index levels at the beginning of the dataset name
- .* (period, single asterisk) at the end of the string represents one or more index levels at the end of the dataset name
- **.* (period, double asterisk, period) in the middle of the string represents either a SINGLE period, or one or more index levels at that point in the dataset name
- The last character specified cannot be a period.
- *** (triple asterisks) are invalid.
- An **.* is invalid, If you want to represent at least a specific number of index levels use **.
Masking Examples

Below is a sampling of some of the dataset name variations that can be built with the masking capability.

- **ABC.GLEDGER.MASTER**  This example is for a complete dataset name
- **XYZ.PROD.GROUP*.MSTR.MUPD**  This example allows multiple for datasets with additional characters as part of the GROUP index level. For example this would pick up GROUP0, GROUP1A, GROUP001, GROUPABC, etc...
- **XYZ.PROD.GROUP++.MSTR.MUPD**  In this variation the GROUP must be followed by 2 numeric digits, such as GROUP25, or GROUP99, etc...
- **PROD.IAM.PAYROLL.*.**  Includes all datasets that begin with
- **PROD.IAM.PAYROLL.*.**  Includes all datasets that begin with PROD.IAM.PAYROLL.
- **XYZ.*.SHARED**  Allows datasets that start with XYZ, have any second level index, and ending with ‘.SHARED,’ for example: XYZ.F1.SHARED or XYZ.ABC12345.SHARED
- **PROD//.ARMASTER.***  Includes datasets that begin with PROD followed by any two valid characters such as PRODNY.ARMMASTER.abc and followed by any characters at the end. Other possible matches include: PRODNJ.ARMMASTER.d060510.master, or PROD99.ARMMASTER.xyz
- **TEST**  Any datasets that begin with the characters TEST.

RLS DSN Table Example

Shown below is an example of an RLS Data Set Name table:

```
SELECT RLSID=RLS1,DSN=(PROD.PAYROLL.IAM.*, PROD.AR///.IAM.*, IAMV.**.RLS, PROD.APAY.*.SHARED)
SELECT RLSID=RLS2,DSN=(PROD2.PAYROLL.IAM.*, PROD2.AR*.IAM.*, IAMV2.*.RLS)
EXCLUDE DSN=(PROD.PAYROLL.IAM.TESTFILE, PROD.AR.IAM.HISTORY, TEST.*, IAMV.RLS999.DONOT.SHARE.*)
```

*Figure 20-6: Example of IAM/RLS Data Set Name Table (EX2020A)*
During OPEN processing of any loaded enhanced format IAM data set, first the dataset is checked to see if the RLSID IAM CREATE Override was specified when the dataset was defined or loaded, and if so then IAM will look for the specified IAM/RLS address space. If the dataset was not defined with an RLSID, next check is if an ACCESS override with RLSID= and which RLS was specified. If both are true, then the specified RLSID will be used for the dataset. Next, the RLS Global Option is checked. If RLS=NONE is in effect, then normal open processing continues. If the RLS Global Option so indicates, the share options are checked. Then based on the RLS Global Options, if the criteria include the dataset name tables, the data set name tables are searched for a match. When IAM refers to the data set name tables during open processing, first the exclude list is searched. If a matching name or mask is found in the exclude table, the IAM data set is treated as being ineligible for IAM/RLS processing. If no match was found in the exclude table, then the selection table is searched. If the data set name or prefix is found in the select table, then the data set is eligible for IAM/RLS processing.

For the preliminary testing phase of IAM/RLS, the installation will most likely set a very limited scope. It may be useful to set the RLS option to a value of NONE, and use the IAM/RLS override or the IAM CREATE RLSID override to initiate processing, or provide a very limited set of eligible data sets.

Subsequent to testing, a wider scope is recommended. Data sets that have shared update access by multiple jobs should be handled by the IAM/RLS address space. Such data sets will normally be defined with share options of (3,3), (3,4), (4,3) or (4,4). These data sets will be automatically routed to the IAM/RLS address space when RLS=SHARE3, if it is active.

In some cases, it may also be beneficial for Share Option 2 data sets to be processed by IAM/RLS. This would be particularly true if an application requires full read integrity while other jobs or CICS regions are updating the data set. If there are only a few share option 2 data sets meeting those criteria, then using the data set name tables would be recommended. However, if almost all of the share option 2 data sets need full read integrity, then it would most likely be better to set the RLS global option to SHARE2.

IAM/RLS does incur additional resources, primarily CPU time, to process an I/O request than without IAM/RLS. It takes additional resources to ship an I/O request over to another address space for processing, and then return data and status back to the originating address space. For this reason, it is recommended that you only put the IAM files that really need sharing support under IAM/RLS, and let standard IAM process the data sets that do not have update sharing active.
IAM/RLS Journaling

Overview

IAM/RLS provides an optional facility to journal before and / or after images of file updates to IAM files being processed by IAM/RLS. These journals can be used to aid in data recovery due to either hardware or software failures. For example, if a media failure occurs and an IAM data set has to be restored, the IAM/RLS journals can be used to reapply the updates that had been made to the file since it was backed up to bring the file status up to the point of failure. If before images are captured then IAM can backout updates performed by failing job steps. The IAM journaling is not a replacement for CICS journaling and CICS dynamic transaction backout, but rather an additional facility that provides for increased recovery capabilities. IAM/RLS as a default, will not journal the before images for updates being done by CICS transactions, because CICS will handle the transaction back outs. However, this can be changed if an installation needs to have those records included in the IAM journals. Many users have found the journal records useful for activity review and auditing purposes.

IAM provides programs to perform recoveries, IAMJREST and IAMBREST. IAMJREST runs as a batch job that can perform forward or backout recoveries. IAMBREST provides for dynamic job step backout of updates performed by a batch job step that abends. IAM also provides a program to perform utility functions on the IAM/RLS journals, called IAMJUTIL. IAMJREST and IAMJUTIL can also be used to process journals created by IAM outside of IAM/RLS control, as described in Section 10.88 IAM Journal and Recovery. Customers can provide their own utility program to process the data contained in the journals, and to even perform recoveries if so desired. For information on the format of the data contained in the IAM/RLS journal records, contact Innovation.

Journaling can be done to either sequential data sets, or using the z/OS system logger. Innovation recommends that customers use the sequential data sets for journaling, unless they have experience in using the z/OS System Logger facilities. The type of journaling and the journal files are specified by the IAM/RLS startup parameters LOGDSNn, SYSLOGGER, and LOGSTRM, which are described in Section 20.10.

Customers select the IAM data sets that they want to have IAM journal, by one of the following methods:

- Specification of the IAM ACCESS Override JRNAD on the IAM/RLS startup procedure, for example: ACCESS DD=&ALLDD,JRNAD=BOTH.
- Specification of the IAM CREATE Override JRNAD when defining the IAM data set(s) that are to have journaling.
- Specification of the LOG parameter on the IDCAMS DEFINE for DFSMS managed IAM data sets.
Sequential Log Data Sets

The sequential log data sets must have space allocated prior to starting IAM/RLS, and are identified by the LOGDSNn= parameter, where n is a numerical digit from 1 to 6. It is recommended that you allocate at least 2 log data sets, preferably 3 or more. Having several large journal data sets will provide for fast access to journal data for job backout recoveries, and help to make sure that there will always be an available journal data set for IAM/RLS, in case some of the journal offload(s) take longer than anticipated. The amount of DASD space required depends on the update activity and the journaling options specified. Each journal record has a header of 112 bytes that is followed by an image of the data record, and is written out in a standard blocked variable length record format. The log data sets may be allocated with secondary space specified, however it is recommended that each log data set reside on a single volume.

JCL to ALLOCATE JOURNALS

Shown below is an example of allocating 6 journal datasets for IAM/RLS use.

```
//ALLOCJRN EXEC PGM=IEFBR14
//JRN1 DD DSN=IAM/RLS.CPUA.JOURNAL1,DISP=(,CATLG),
  //   UNIT=SYSDA,VOL=SER=JRNVL1,SPACE=(CYL,(1000))
//JRN2 DD DSN=IAM/RLS.CPUA.JOURNAL2,DISP=(,CATLG),
  //   UNIT=SYSDA,VOL=SER=JRNVL2,SPACE=(CYL,(1000))
//JRN3 DD DSN=IAM/RLS.CPUA.JOURNAL3,DISP=(,CATLG),
  //   UNIT=SYSDA,VOL=SER=JRNVL3,SPACE=(CYL,(1000))
//JRN4 DD DSN=IAM/RLS.CPUA.JOURNAL4,DISP=(,CATLG),
  //   UNIT=SYSDA,VOL=SER=JRNVL4,SPACE=(CYL,(1000))
//JRN5 DD DSN=IAM/RLS.CPUA.JOURNAL5,DISP=(,CATLG),
  //   UNIT=SYSDA,VOL=SER=JRNVL5,SPACE=(CYL,(1000))
//JRN6 DD DSN=IAM/RLS.CPUA.JOURNAL6,DISP=(,CATLG),
  //   UNIT=SYSDA,VOL=SER=JRNVL6,SPACE=(CYL,(1000))
```

Figure 20-7: Example of JCL to allocate IAM/RLS Journals (EX2030A)
When IAM/RLS is started, the journal subtask will look at all specified journal data sets. It will select the one with the most recent data (as determined by timestamps within the journal records). If all of the journal data sets are empty, it will start with LOGDSN1. As each journal data set is filled, or manually switched with the JSWITCH command, IAM/RLS will use the next higher LOGDSNn data set. When the highest one has been used, then IAM/RLS will switch back to the first one, LOGDSN1.

LOGDSN1=IAMRLS.CPUA.JOURNAL1
LOGDSN2=IAMRLS.CPUA.JOURNAL2
LOGDSN3=IAMRLS.CPUA.JOURNAL3
LOGDSN4=IAMRLS.CPUA.JOURNAL4
LOGDSN5=IAMRLS.CPUA.JOURNAL5
LOGDSN6=IAMRLS.CPUA.JOURNAL6
JRNPROC=IAMJRNP1
SYSLOGGER=NO

Figure 20-8: Example of IAM/RLS Startup parameters for Sequential Journals (See EX2010A)

When IAM/RLS performs a journal switch, it will start the procedure identified by JRNPROC to offload the journal and either empty the data set or flag it for reuse. If JRNPROC has not been specified, the customer must make sure that the journal data set is offloaded and emptied prior to IAM/RLS requiring the data set again. Users that have automated operations software can trigger a job or procedure by watching for the IAML0510 message, indicating that a journal data set has been closed. If a journal switch occurs, and the next journal data set is not ready to be written to, then IAM/RLS will issue a WTOR for operator interaction on how to proceed, which will delay IAM/RLS processing.

Data from an IAM/RLS journal can be offloaded by using the IAMJUTIL program, by IEBGENER, or with their own utility program. Innovation recommends the use of IAMJUTIL because it offers selection criteria on which journal records are copied, and it can indicate that the journal data set can be reused, while retaining the data in the journal data set until it is reused. The advantage of reuse is that it simplifies job backout recoveries, because a lot more of the journal data will be readily available than if the journal had been emptied. If you are planning on using the Dynamic Job Backout function of IAM/RLS, then you definitely should be using IAMJUTIL to offload the journal data sets.

One scenario for managing the journals is when a journal is full, to use IAMJUTIL to copy only the after images to a tape, which can be used for subsequent forward recovery if needed. The before images, used for backout recovery will be discarded upon reuse of a journal data set. By providing a large enough set of journals such that the immediate backout needs can be met, this will reduce the amount of journal data that has to be saved.
An example of a "proc" that can be started by IAM/RLS, which will copy the after images to a GDG is:

```plaintext
//IAMJRNP1 PROC LOG=NULLFILE
//*******************************************************
//--COPY AFTER IMAGES FROM IAM JOURNAL TO A GDG, AND
//--MARK JOURNAL AS ELIGIBLE FOR REUSE BY IAM/RLS
//--*******************************************************
//IAMJUTIX EXEC PGM=IAMJUTIL
//SYSUDUMP DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//IAMJRNL DD DISP=SHR,DSN=&LOG  
//JRNLOUT DD DISP=(,CATLG),DSN=gdg(+1),UNIT=TAPE  
//SYSOUT   DD SYSOUT=*  
//SYSIN    DD DISP=SHR,DSN=IAMSYS.RLS.PARMLIB(COPYAFTR)

Contents of IAMSYS.RLS.PARMLIB(COPYAFTR):

COPY AFTER,REUSE
```

*Figure 20-9: Example of PROC to Offload Journal After Image Records (EX2030B)*

If you will not be making use of the IAM Dynamic Job Backout capability, then other alternative procedures could be used. For example, once a journal data set is filled, it could be renamed, and space allocated then for the original data set name. Or, you could have a secondary set of journals, and use the IAM/RLS CHANGEParm command to switch the journaling to the alternate set.
System Logger

As an alternative to the traditional sequential log data sets, customers can optionally use the z/OS System Logger DASD-only logging instead. IAM/RLS can use the z/OS system logger for journaling file updates to allow for back out and forward recovery of IAM files. The z/OS system logger is a component of z/OS that must be activated and configured. For more information on the z/OS system logger see the following publications:

- z/OS MVS Setting Up A Sysplex
- z/OS MVS Initialization and Tuning Reference

You must perform the following tasks in order to set up the IAM System Logger environment. Your z/OS systems programmer must do some of these tasks.

1. Make sure you have authorization to the z/OS system logger address space and that the z/OS system logger (IXGLOGR) is running before you define and use IAM log streams. The section "Define Authorization to System Logger address space" in the z/OS MVS Setting Up a Sysplex manual provides more information about this.

2. You must also have authorization to the MVS IXCMIA PU utility. This utility is used to define, update, and delete entries in the LOGR couple data set. IXCMIA PU is documented in the z/OS MVS Setting Up a Sysplex manual in appendix B.

3. Use the IXCL1DSU utility to define and format the LOGR couple data set. Your installation may have already done this if they are using the z/OS system logger for other products. This utility is documented in the Z/OS MVS Setting Up a Sysplex manual.

4. Define the IAM log streams in the LOGR couple data set using the z/OS utility IXCMIA PU. See the section "Add Information about Log Streams and Coupling Facility Structures to the LOGR Policy" in the z/OS MVS Setting Up a Sysplex manual. The following JCL can be used to set the values required for the IAM log stream using the IXCMIA PU utility.
Shown below is an example of the JCL to define the log stream.

```plaintext
//STEP1 EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSPRIOUT=*  
//SYSIN DD *  
  DATA TYPE(LOGR) REPORT(YES)  
  DEFINE LOGSTREAM NAME(IAMLOGR.CPUB)  
    DESCRIPTION(IAM)  
    DASDONLY(YES)  
    STG_STORCLAS(TMPDATA)  
    LS_STORCLAS(TMPDATA)  
    STG_SIZE(48000)  
    LS_SIZE(48000)  
    HLQ(SYSPXCF)  
    LOWOFFLOAD(0)  
    HIGHOFFLOAD(50)  
    DIAG(YES)  
/*
```

**Handling Errors**

When there is an error, the IAM/RLS JRNAD exit will end up waiting on the buffer queue until buffers are emptied and the request on top of the queue gets posted when the error is resolved. At this time the only error automatically handled is for reason code of 00000860 indicating that an offload is needed. All other tasks trying to journal are stacked up behind the one on the buffer queue waiting for the journal buffer lock.

What can be done if there is a catastrophic error with the logstream, or coupling facility is the following:

1. Disconnect from the current logstream: `F iams1,LOGGER,DISCON`
2. Define a new logstream
3. Change the IAM/RLS parameter library LOGSTREAM value.
4. Issue the IAM/RLS CHANGEPARM command: `F iams1s1,CHANGEPARM`
5. Reconnect to the new logstream: `F iams1,LOGGER,RECON`
6. At that point the waiting request should be posted. If not, then issue the IAM/RLS LOGGER POST command: `F iams1,LOGGER,POST`
20.40 IAM/RLS Operator Commands

Certain aspects of the IAM/RLS address space can be controlled by an operator by the
z/OS modify (F) command. Some of the functions include shutting down the IAM/RLS
address space, causing the IAM/RLS address space to schedule a dump to a
SYS1.DUMPxx data set, activating and stopping the IAM/RLS trace facility, and
applying PTF’s to an active IAM/RLS address space. The general format of the
command is as follows:

```
F IAMRLS address space name,command
```

For example, to dump an IAM/RLS address space started with the name of IAM/RLS,
enter the following:

```
F IAMRLS,DUMP
```

The following IAM/RLS commands can be used. Please note that some of the
commands will prompt you to enter additional information via a WTOR. Also note that
where a recognized abbreviation exists for a command or for operand it is shown by the
underlined portion of the full value.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLY,dsn</td>
<td>This command is used to initiate application of Innovation supplied PTF’s to the modules loaded in the IAM/RLS address space storage. This facility enables customers to apply most maintenance without stopping and restarting IAM/RLS processing. For dsn, specify the name of a sequential or partitioned dataset containing the fix to be applied. (For PDS’s, include the member name in parenthesis.) For example:</td>
</tr>
<tr>
<td>CHANGEDSNT</td>
<td>This command will update the list of dataset names to be included and/or excluded from IAM/RLS processing from the IAMDSNTB DD statement. This provides customers with a mechanism for picking up an updated list without having to stop and restart IAM/RLS.</td>
</tr>
<tr>
<td>CHANGEPRAM</td>
<td>This command will update the current IAM/RLS parameters from the IPARMLIB DD statement file. While most parameters can be dynamically changed, some parameters will only effectively be changed by a restart of IAM/RLS. For example, because MAXLOCKS is used to set an internal table size established at startup, changing that value will not have any effect until IAM/RLS is restarted.</td>
</tr>
</tbody>
</table>
**CLOSEFILE**

This command will close the specified data set. It is intended to be used only in emergency circumstances to recover from a problem. Datasets are normally closed by the applications using IAM/RLS. The file should first be closed by the application or CICS region before attempting to issue the Closefile command. Additional operands, separated by commas, are:

- **DSN=** Specifies the data set name.
- One of the following is also required:
  - **JOBNAME=** Specifies the jobname for which the dataset is to be closed.
  - **JOBID=** Specifies the JES JOBID that identifies the job for which the data set is to be closed.
  - **ALL** Specifies that the file is to be closed for every job that has it opened under IAM/RLS.

**DISPLAY**

This command, along with associated subparameters, will display requested information on the output file RLSLOGDD. The RLSLOGDD is used as a default because the output from the DISPLAY could be quite substantial. By adding the keyword ',CON' to the end of the DISPLAY request, the information will be displayed on the operator's console. One of the following parameters must be provided to indicate what information is to be displayed.

- **CONTENTION** – Requests IAM/RLS to display any record locks held that have contention (other requestors waiting for the lock).
- **EXCLUDETB** – Indicates that the data set name exclusion table is to be displayed.
- **RETAINEDLOCKS[,DSN]** – Indicates that information on the jobs, and optionally which data sets used by those jobs, have had locks retained due to an abend will be displayed.
- **SELECTTB** – Indicates that the data set name selection table is to be displayed.

For example: **F IAMRLS,DISPLAY,RETAINEDLOCKS**

**DJBTRACE=**

This command is used to activate or deactivate the Dynamic Job Backout tracing. The Dynamic Job Backout trace is useful for determining exactly what file activity is being performed by Dynamic Job Backout. This can be used to verify proper operation, or for problem diagnosis relating to the recovery processing. When the trace is activated, by specifying DJBTRACE=YES, the next time a backout is attempted, the backout will dynamically allocate the RLSDJBDD file to a SYSOUT data type of dataset. When deactivated, by DJBTRACE=NO, upon completion of a backout, the output for the trace will be closed and deallocated. Valid values are:

- **YES** – Activate IAM/RLS Dynamic Job Backout tracing.
- **NO** – Deactivate IAM/RLS Dynamic Job Backout tracing.

**DUMP**

This command is used to request that the IAM/RLS address space take a dump to a SYS1.DUMPxx data set. The dump will automatically include the data space containing the IAM Index Space.
**JSWITCH**
This command causes logging to switch from the currently active log data set to the next log data set.

**LOGSWITCH**
This command causes IAM/RLS to close and deallocate the currently active file being used for output directed for the IAM/RLS log (RLSLOGDD). Another SYSOUT dataset will be dynamically allocated using the SYSOUT class specified in the IAM/RLS startup parameters. If an actual RLSLOGDD DD card is in the IAM/RLS startup proc, then this command is ignored.

**LOGGER**
These are some commands to help deal with any problems with the System Logger. What can be done here is that if there is a catastrophic error with the logstream, or coupling facility, they can disconnect from the current logstream, define a new logstream, change the RLS parmib, run the changeparms command, then reconnect to the new logstream. At that point BJRNL should get posted by the SRB to recover, but just in case they can issue the POST command to get things started.

There are 4 different parameters:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISCON</td>
<td>Used to disconnect from the current System Logger logstream</td>
</tr>
<tr>
<td>RECON</td>
<td>Used to reconnect to the System Logger logstream</td>
</tr>
<tr>
<td>OFFLOAD</td>
<td>Used to start an offload for the current logstream</td>
</tr>
<tr>
<td>POST</td>
<td>Post the journal subtask to start writing log records again</td>
</tr>
</tbody>
</table>

For Example: F iamplex1.OFFLOAD

**MESSAGES=**
Specifies what messages should be printed to the IAM/RLS RLSLOGDD file. Valid values are ALL, INFO, or ERR. ALL is primarily intended for problem diagnosis, as various messages will be written indicating many frequent events. INFO will provide a fairly detailed activity log, including when data sets are opened and closed. ERR will limit the RLSLOGDD messages to only those issued in error situations.

**QUIESCE[,FORCE]**
This command requests that the IAM/RLS address space shut down. If FORCE is not specified, then when all of the currently open IAM files are closed, the IAM/RLS address space will terminate. Any new open requests will be rejected.

If the FORCE keyword is specified, then all of the files will be immediately closed, and the address space will terminate normally. All subsequent requests by the jobs that were processing data sets through the IAM/RLS address space will be rejected.
### IAM/RLS Operator Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| RELEASELOCKS  | Specifies that IAM/RLS is to release locks for the specified job or CICS transaction. Either retained or held locks may be released. Additional parameters may also be required to further identify the specific job step if there are multiple matches on the job name. The information necessary to provide the details required by this command can be obtained from the DISPLAY, RETAINEDLOCKS command. The following sub-parameters can be specified:  
  - **JOBNAME**= specifies the 1 to 8 character job name.  
  - **JOBID**= Specifies the 8 character JES jobid, for example: JOB12345  
  - **ASID**= Specifies the hexadecimal digits of the address space ID number.  
  - **STEPNAME**= Specifies the 1 to 8 character step name, and is optional.  
  - **TRNAME**= Specifies the 1 to 4 character CICS transaction name.  
  - **TRID**= Specifies the CICS transaction number, in 1 to 5 decimal digits. |
| RESTART       | This command will reverse a previously issued QUIESCE command that was issued without the FORCE keyword. Normal processing will be resumed by the IAM/RLS address space, and new open requests will be honored. This will only function if the IAM/RLS address space has not already terminated. |
| RESTORE,dsn    | This command can be used to reverse the application of an Innovation supplied PTF from the modules loaded within the IAM address space. This can be used when a PTF has been applied using the APPLY command, and is either causing additional problems, or if it did not resolve the problem. For **dsn** specify the same value as was provided on the APPLY command. |
| TRACE         | This command activates the IAM/RLS internal tracing. Refer to the Section 20.41 on IAM/RLS tracing for additional information on using the trace facility. |
IAM/RLS Tracing Facility

The IAM/RLS Trace Facility enables Innovation support personnel to request information about I/O requests and other processes that are occurring within the IAM/RLS Address Space. Trace output will be to dynamically allocated SYSOUT spool datasets or to previously allocated output files. All trace requests are initiated via the MVS console MODIFY (F) command. If additional parameters are required beyond what was initially specified, then the trace processor will request them via a WTOR. For trace start requests, the additional parameters can alternatively come from a sequential data set. The general format of using the IAM/RLS trace command is as follows:

```
F IAMRLS,TRACE,subcommand,parameters
```

The “IAM/RLS” is replaced by the IAM/RLS address space name. The “subcommand” are one of the subcommands shown below. The parameters will be as indicated for the subcommand. There are three IAM trace subcommands for IAM/RLS and all are entered via the MVS MODIFY command. The subcommands are:

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>Request a new trace</td>
</tr>
<tr>
<td>STOP</td>
<td>Stop an existing trace</td>
</tr>
<tr>
<td>LIST</td>
<td>List all active trace requests</td>
</tr>
</tbody>
</table>

Figure 20-11: Trace subcommands

Starting a Trace

To start a trace, issue the command:

```
F IAMRLS,TRACE,START
```

The additional parameters can be included on the above command, provided via replies to the WTOR’s, or with the PARMS= parameter, through a sequential data set. Parameters for TRACE START subcommand are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSN=</td>
<td>Optional parameter that specifies the name of the IAM dataset that you wish to trace requests against. Trace information will be generated by any request from any address space that is accessing the named file unless further limited by JOBNAME= and/or STEPNAME= parameters.</td>
</tr>
<tr>
<td>END</td>
<td>Required parameter to signify that all of the desired parameters had been specified.</td>
</tr>
<tr>
<td>ID=</td>
<td>Optional parameter that specifies a unique 1 to 8 alphanumeric character string used to identify this trace request. If one is not specified, IAM will generate a value. This ID is required for STOP and LIST requests.</td>
</tr>
<tr>
<td>INTERNAL</td>
<td>Specifies that trace will be done to an internal wrapping table in 64-bit virtual storage. The table size will be 2 megabytes. This is primarily intended for gathering additional information as may be requested to diagnose a problem.</td>
</tr>
<tr>
<td>JOBNAME=</td>
<td>Optional parameter that specifies tracing is to be performed for requests from specific job. If not specified, all jobs with IAM data sets that are accessed through the IAM/RLS address space will be eligible for tracing. Use this parameter with DSN= and/or STEPNAME= to limit what is traced.</td>
</tr>
<tr>
<td>JOB=</td>
<td></td>
</tr>
</tbody>
</table>

Figure 20-12: Trace parameters
Stopping a Trace

To terminate tracing, issue the following command:

```
F IAMRLS,TRACE,STOP,ID=traceid,END
```

The parameters for TRACE STOP requests are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID=</td>
<td>Required parameter that specifies the ID of the TRACE request to be stopped.</td>
</tr>
<tr>
<td>END</td>
<td>Required to signify that all parameters have been specified.</td>
</tr>
</tbody>
</table>
Displaying Active Trace Requests

Use the TRACE,LIST command to display the currently active trace requests. You can request either all of the currently active trace requests, or a specific trace request.

The parameters for TRACE,LIST requests are:

Parameter:  Description:
ID=          Optional parameter that specifies the trace ID from the TRACE,START request. If not specified, all active trace requests will be displayed on the MVS console.
END          Required parameter to indicate that all of the desired parameters have been specified.

Examples of Trace Commands

The following demonstrate some examples of using the IAM/RLS trace function. All examples assume an IAM/RLS address space job name of IAM/RLS.

To trace all I/O Start and I/O End requests from all IAM files from CICS region CICDIDP1:

F IAMRLS,TRACE,START,ID=IAM1,JOBNANME=CICSIDP1,TRACEREQ=(IOS,IOE),END

To trace all PC calls against dataset MY.IAMFILE:

F IAMRLS,TRACE,START,ID=PCTRACE,DSN=MY.IAMFILE,TRACEREQ=PC,END

To list all active trace requests:

F IAMRLS,TRACE,LIST,END

To stop a trace request with the ID of TRACE1:

F IAMRLS,TRACE,STOP,ID=TRACE1,END
IAM/RLS CICS Considerations

Overview

CICS TS (Transaction Server) customers can use IAM files under IAM/RLS, without changes to their application programs or transactions. (Note: IAM/RLS cannot be used for releases prior to CICS/TS). Additionally, there are no changes required to the IAM file definitions under CICS. There are a couple of IAM provided CICS exit routines that must be installed and activated to use IAM/RLS under CICS. These exits are needed so IAM/RLS can provide the required record locking protocols. Exits are used for IAM/RLS to recognize when a logical unit of work (UOW) completes due to a SYNCPOINT request, or transaction termination, which is done using the CICS Task Related User Exit (TRUE) facility. IAM also provides a CICS Global User Exit (GLUE) for backout to recognize when a transaction backout is taking place that is subsequent to the original transaction, such as can occur during an emergency restart. If these exits are not activated, then attempts to OPEN IAM files for IAM/RLS processing under CICS will be failed.

There is an additional Global User Exit (GLUE) that is provided and must be installed if it is desired that IAM abend any transactions that have encountered lock contention, and have indicated that IAM/RLS is to ABEND such transactions.

IAM/RLS CICS Exits

IAM customers that wish to use the IAM/RLS functionality in their CICS regions must install and activate the provided Task Related User Exit (TRUE), IAMBCICS, the provided Global User Exit (GLUE), IAMXFCBO, IAMXFCRQ and the CICS transaction program IAMXCNTL. To do this, the following steps are required.

- Insure that modules IAMXCINI, IAMBCICS, IAMXFCBO, IAMXFCRQ and IAMXCNTL are in a data set that is part of the DFHRPL concatenation in the CICS JCL.

- Insure that modules IAMXCINI, IAMBCICS, IAMXFCBO, IAMXFCRQ and IAMXCNTL are defined to CICS in DFHCSD.

- The four modules IAMXCINI, IAMBCICS, and IAMXFCBO, IAMXFCRQ must be defined with the following attributes:
  - LANGUAGE=ASSEMBLER
  - RELOAD=NO
  - DATALOCATION=ANY
  - EXECKEY=CICS.

- The module IAMXCNTL must be defined with the following attributes:
  - LANGUAGE=ASSEMBLER
  - RELOAD=NO
  - DATALOCATION=ANY
  - API=OPENAPI
  - EXECKEY=CICS.

- Alternatively, one can use the CICS batch utility program DFHCSDUP to define the IAM exits. The following JCL example shows how to run the CICS DFHCSDUP program to define the IAM exits to a specified CICS CSD (CICS System Definition file). The input as specified by the SYSIN DD card is the ICL (IAM Installation Control Library) member IAMEXIT5 which should be reviewed prior to running the update job shown below.
IAM/RLS and CICS Initialization

Using IAM/RLS and CICS requires that the IAM exits are defined and enabled during both the stage II and the stage III initialization phases of CICS. The IAM program IAMXCINI performs this initialization of the necessary exits and additional internal storage areas for IAM and must be added to the CICS PLTPI program table. The program IAMXCINI must be added to the second and third stage of the PLTPI list that is used at CICS startup. An example of how this might look is:

```
DFHPLT TYPE=INITIAL,SUFFIX=X1
DFHPLT TYPE=ENTRY,PROGRAM=IAMXCINI For CICS/TS 3.1+
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
DFHPLT TYPE=ENTRY,PROGRAM=IAMXCINI
DFHPLT TYPE=FINAL
END
```

The IAM program IAMXCINI, in the PLTPI functions as an exit activator within CICS to initialize the IAM and CICS environment. It will programmatically issue the following CICS SPI (System Program interface) commands:
IAM/RLS CICS Considerations

- EXEC CICS ENABLE PROGRAM(IAMXFCBO) EXIT(XFCBOUT) START
- EXEC CICS ENABLE PROGRAM(IAMXFCRQ) EXIT(XFCREQ) START
- EXEC CICS ENABLE PROGRAM(IAMXFCRQ) EXIT(XFCREQC) START
- EXEC CICS ENABLE PROGRAM(IAMBCICS) TASKSTART START

These commands activate the CICS customization exits at GLUE and TRUE points respectively for required IAM/RLS functionality within the CICS region.

During CICS Initialization stages, IAM will issue to the CICS job log the following console messages:

CICS PLT Stage II messages:

DFHSI8420I CICSB319 About to link to PLT programs during the second stage of initialization.
IAML1003 IAMXCINI - IAM PLT Stage II initialization has begun for IAM v9.1/00A.
IAML1004 IAMXCINI - IAM TRUE exit program: IAMBCICS has been ENABLED.
IAML1005 IAMXCINI - IAM GLUE exit program: IAMXFCBO has been ENABLED at EXIT(XFCBOUT).
IAML1005 IAMXCINI - IAM GLUE exit program: IAMXFCRQ has been ENABLED at EXIT(XFCREQ).
IAML1005 IAMXCINI - IAM GLUE exit program: IAMXFCRQ has been ENABLED at EXIT(XFCREQC).
IAML1009 IAMXCINI - IAM Initialization ENQUEUE was issued.
IAML1006 IAMXCINI - IAM GLUE exit program: IAMXFCBO has been STARTED at EXIT(XFCBOUT).
IAML1006 IAMXCINI - IAM GLUE exit program: IAMXFCRQ has been STARTED at EXIT(XFCREQ and XFCREQC).
IAML1003 IAMXCINI - IAM PLT Stage II initialization has ended.
DFHSI8424I CICSB319 Control returned from PLT programs during the second stage of initialization.

CICS PLT Stage III messages:

DFHSI8430I CICSB319 About to link to PLT programs during the third stage of initialization.
IAML1003 IAMXCINI - IAM PLT Stage III initialization has begun for IAM v9.1/00A.
IAML1011 IAMXCINI - IAM program: IAMXCINI was loaded at x(8B440028).
IAML1011 IAMXCINI - IAM program: IAMBCICS was loaded at x(800A1000).
IAML1011 IAMXCINI - IAM program: IAMXFCBO was loaded at x(8B446028).
IAML1011 IAMXCINI - IAM program: IAMXFCRQ was loaded at x(8B444BD8).
IAML1011 IAMXCINI - IAM program: IAMXFCAB was loaded at x(90E5FA00).
IAML1011 IAMXCINI - IAM program: IAMXFCRQ was loaded at x(90E5FA00).
IAML1012 IAMXCINI - IAM program: IAMXFCRQ was loaded at x(90E5FA00).
IAML1012 IAMXCINI - IAM program: IAMXFCAT was loaded at x(8CC00028).
IAML1007 IAMXCINI - IAM TRUE exit program: IAMBCICS has been STARTED.
IAML1003 IAMXCINI - IAM PLT Stage III initialization has ended.
DFHSI8434I CICSB319 Control returned from PLT programs during the third stage of initialization.

The very first IAM file that is shared via IAM/RLS will cause at OPEN time, the following console messages to occur:

+IAML1000 IAMRLS/CICS Initialization complete. IAM RLS Support active v9.2/00 A (000A1000)
+IAML1001 IAMRLS/CICS is connected to IAM RLS with RLSID = RLS1
CICS Automatic Disconnect and Reconnect after a restart of IAM/RLS

Within a CICS region, IAM has the capability to automatically disconnect and then subsequently reconnect to an IAM/RLS address space after the IAM/RLS address space has been terminated and then has restarted. This eliminates the need to restart or recycle one or multiple CICS regions that are sharing IAM datasets via the IAM/RLS option.

The IAM code that does the setting of CICS FILEIDs closed, attempting the RETRY of shunted UOWs and then setting the FILEIDs OPEN and AVAILABLE, runs underneath the default transaction ID of $IAM, unless a different transaction ID has been specified in the IAM Global Options for the CICSTXID= operand. The transaction will run as an CICS OPENAPI subtask so as to facilitate the speedy disconnect and reconnects and closing and opening of IAM files affected by an IAM/RLS instance being terminated and restarted.

Disconnect Process

An IAM/RLS address space can terminate for a variety of reasons such as an operator issued a normal IAM/RLS QUIESCE command or an MVS "P" (stop or halt) command to the IAM/RLS instance or due to an abnormal termination or cancelling of the IAM/RLS instance. In either situation, IAM code within the CICS region determines that the IAM/RLS region to which it is connected has terminated. Initially the IAM code in CICS will build and maintain a list of the IAM files that were opened in CICS and being shared via the affected IAM/RLS address space. While IAM is building this list of affected IAM files within the CICS region, IAM will attempt to set each of these IAM files to a CICS "CLOSED" state when there are no active UOWs (unit-of-work) for the IAM file.

However if there are active UOWs for an affected IAM file, then internally IAM code will issue a CICS "FORCECLOSE" command. This "FORCECLOSE" command will cause any CICS task(s) that have active UOWs to the IAM file to attempt to go through normal CICS backout. That is they will be "rolled back" (any updates in flight will be backed out). However, as CICS will see the IAM file as being in a "not available" status (due to the IAM/RLS address space instance having been terminated) CICS will cause the UOWs to be retained, logged, and converted into "shunted UOWs". These "shunted UOWs" will later get resolved when the affected IAM file becomes available again.

Disconnect Messages

There are a variety of different messages that may be seen in different places when an IAM/RLS instance is not longer active. Below are examples of these messages:

On a CICS terminal for inflight transactions at time of disconnect:

DFHAC2205 13:00:55 CICSB329 A backout failure has occurred during syncpoint processing for transaction XXXX

On the system console and the CICS joblog:

IAML1001 IAMRLS/CICS has disconnected from IAM RLS with RLSID = RLB9 (LOCAL)
On the system console and CICS joblog:

```
IAMW15 DD=KSDS0  REQUEST FAILED - IAM ADDRESS SPACE QUIESCING OR INACTIVE
IAMW15 DD=KSDS0  DSN=IAMCICSV.$IAM.KSDS0
IAMW15 DD=KSDS0  REQUEST FAILED - IAM ADDRESS SPACE QUIESCING OR INACTIVE
IAMW15 DD=KSDS0  DSN=IAMCICSV.$IAM.KSDS0
DFHFC4700 CICSB329  832
A VSAM error has occurred during file backout. (Module DFHFCFR has
 returned reason code (X'59'), access method code (X'08B90000') and
 length error code (X'00').)
DFHMED016 CICSB329  833
(Module:DFHMEME) CICS symptom string for message DFHFC4700 is
 PIDS/655M1500 LVLS/650 MS/DFHFC4700 RIDS/DFHFCRC PTFS/UK73688
 PCSS/REASONCODE PRCS/0000059 PCSS/ACCMETHCOD PCSS/E PRCS/08B90000
 PCSS/LENGERRCOD PCSS/E PRCS/00000000
DFHDEU0205 CICSB329 A SYSTEM DUMP FOR DUMPCODE: FC4700 , WAS SUPPRESSED BY THE
 DUMP TABLE OPTION FOR THIS DUMPCODE
DFHFC4701 CICSB329  835
09/13/2012 13:00:55 CICSB329 Backout failed for transaction CECI,
 VSAM file KSDS0, unit of work X'CA2A87F954E92905', task 00189, base
 IAMCICSV.$IAM.KSDS0, path IAMCICSV.$IAM.KSDS0, failure code X'FE'.
DFHDEU0205 CICSB329 A SYSTEM DUMP FOR DUMPCODE: FC4701 , WAS SUPPRESSED BY THE
 DUMP TABLE OPTION FOR THIS DUMPCODE
```

On the CICS Destination Log CSMT(DD=MSGUSR):

```
IAML1050  Processing of CLSF request has begun for RLSID=RLB9
DFHFC0201 11/16/2012 17:13:59 CICSB329 Non-RLS file KSDS0 has been
deallocated. Module DFHFCN.
IAML1051  Processing of CLSF request has completed for RLSID=RLB9
```

**Reconnect Process**

When the IAM code within a CICS region determines that the IAM/RLS address space has been subsequently restarted, IAM will issue a reconnect message and then using the saved list of IAM files IAM will attempt to reopen each of the affected file. Initially IAM will attempt a SET DATASET RETRY for each of the associated IAM datasets for each of the corresponding CICS FILEIDs in the saved IAM file list. This "RETRY" option will cause any existing "SHUNTED UOWs" to be retried. That is the previously failed or shunted units of work that were in flight at the time that the IAM/RLS address space terminated will be rolled back (backed out). After the "shunted UOWs" have been retried, then IAM causes the affected CICS FILEIDs to be set to an OPEN and ENABLED state. At this point normal processing and access to the IAM files via the connected IAM/RLS region can proceed.
Shown below are examples of messages that can appear on the console and CICS joblog when IAM/RLS has been restarted:

<table>
<thead>
<tr>
<th>IAML1001</th>
<th>IAMLRS/CICS has reconnected to IAM RLS with RLSID = RLB9 (LOCAL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAML2005</td>
<td>IAMXFS1 HAS BEEN POSTED FOR RLSID = RLB9</td>
</tr>
<tr>
<td>IAML2008</td>
<td>IXCQUERY RC=00000000 , RSN= 00000000</td>
</tr>
<tr>
<td>DFHFC0970</td>
<td>CICSB329 Recoverable non-RLS file KSDS0 opened with VSAM SHROPT 3 or 4. CICS cannot ensure integrity.</td>
</tr>
</tbody>
</table>

On the CICS Destination Log CSMT (DD=MSGUSR):

<table>
<thead>
<tr>
<th>IAML1050</th>
<th>Processing of RTRY request has begun for RLSID=RLB9</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFHFC0200</td>
<td>11/16/2012 17:15:41 CICSB329 Non-RLS file KSDS0 has been allocated to data set IAMCICSV.$IAM.KSDS1. Module DFHFCN.</td>
</tr>
<tr>
<td>IAML1064</td>
<td>OPEN completed for FILE(KSDS0) with DSN(IAMCICSV.$IAM.KSDS0) at 17:15:37.</td>
</tr>
<tr>
<td>IAML1051</td>
<td>Processing of RTRY request has completed for RLSID=RLB9</td>
</tr>
</tbody>
</table>
GLUE for Record Lock Timeout

For some IAM/RLS users it may not be practical to deal with the “ILLOGIC” condition raised in CICS when a transaction has timed out while waiting for a record lock. This can occur if a customer has specified that time outs should occur when waiting for locks under IAM/RLS on the CICSLOCK parameter (See Section 20.10). For those users, a program is provided that runs as a CICS Global User Exit (GLUE), which will purge transactions that receive the IAM/RLS specific ILLOGIC condition.

A site wishing to activate this functionality needs to do the following:

- Insure that modules IAMXFCAB and IAMXFCAT are in a data set that is part of the DFHRPL concatenation in the CICS JCL.
- Insure that modules IAMXFCAB and IAMXFCAT are defined to CICS in DFHCSD.
- Add IAMXFCAT to the second stage of the PLTPI list that is used at CICS startup.
- Restart CICS.

The exit activator program, IAMXFCAT, programmatically issues the command “EXEC CICS ENABLE PROGRAM(IAMXFCAB) EXIT(XFCREQC) START”. This command causes CICS to enter IAMXFCAB at the conclusion of each file request made through the standard CICS API before returning control to the application logic.

The exit program itself checks the EIBR CODE for the IAM/RLS deadlock specific indicators, and if they are found it programmatically issues a command to purge the transaction. This results in a transaction abend with a code of AKC3. Users that decide to use this feature will have to determine if they wish to suppress this dump code or use the dumps to identify the files and the activities that are involved in dead lock situations.

Users that have other program(s) enabled for the XFCREQC exit should generally enable IAMXFCAB last, CICS will run programs at exits in the order they were enabled.

CILOCK

The CICS System Initialization Table (SIT) parameter CILOCK defaults to NO. This setting is intended to reduce the number of VSAM exclusive control conflicts when reading VSAM files for update. It does so by using extra CPU cycles and doing extra I/O. IAM locks at the record level, not at the control interval level, thereby eliminating this delay except when concurrent transactions are attempting to update the same records. Therefore the CILOCK parameter should be set to YES when most or all of the VSAM files have been converted to IAM.

If CICS is running with the default CILOCK value of NO and IAM files are being shared with IAM/PLEX or IAM/RLS, then it is critical to provide a sufficient value for the VSAM STRINGS. Failure to do so can result in CICS deadlock situation if a waiting for strings condition occurs.
IAM/RLS Batch Considerations

Overview

The ability to maintain and achieve data integrity can be a complex matter when dealing with non-transactional processing, as is the case with many batch applications. Such batch applications range from performing a relatively simple and straightforward update process on one or only a few files, to very complex processing involving updates to a large number of related datasets and databases. Additionally, such applications may not have been designed or written with record level data sharing as a consideration. While software products, such as IAM/RLS, can support and facilitate data sharing, there may be a need to change some batch applications to facilitate good performance and data integrity in a data sharing environment.

The discussion of IAM/RLS and batch processing below is intended to provide assistance in determining what, if any, changes may be necessary to the batch job streams that will be processing files managed by IAM/RLS. The main considerations that should be kept in mind are what the recovery requirements for the files being processed by the batch jobs are and what other concurrent activity is likely to be occurring on the shared files while the batch job(s) are executing. One of the key points is that with concurrent updating going on, you cannot just restore all of the files that were updated by the batch job and rerun it, because updates made by the other concurrent processing will then be lost. Different recovery techniques will have to be used.

IAM/RLS and Batch Jobs

IAM/RLS can be used with batch jobs that read and update IAM files. With IAM/RLS automatic dataset selection, there are no changes required to JCL for batch jobs to use IAM files under IAM/RLS. The main area that requires some consideration is for those batch jobs that update recoverable IAM files with IAM/RLS. For purposes of this discussion a recoverable IAM file is one that has before image journaling. Because such files can have updates backed out should the batch job fail the record locks are retained until the batch job step ends normally, until a batch syncpoint call is performed or if the job step fails until a recovery is completed.

A job step failure from the IAM/RLS viewpoint would be a job step abending. IAM/RLS does not look at application return codes, or the condition codes set for a particular job step. If the job step abends, it is considered a failure. If a job step terminates without abending, then IAM/RLS considers it successful. This is significant because if a recovery is needed, a valid recovery would consist of only recovering to either the last syncpoint if there is one or to the beginning of the job step if no syncpoints were taken. Attempting to backout updates performed by prior job steps may cause a loss of data, because other jobs or online systems may have updated those records subsequent to their completion. Those updates would be lost if a recovery is performed for job steps other than the failing job step. Likewise, attempting to backout updates for a job step the application considers failed due to condition codes could also result in data loss. Backouts of all of the updates made by the failing job could only be performed if the only update access was being performed by the failing job. This can only be guaranteed if the failing job allocates the IAM dataset(s) with DISP=OLD. Otherwise, there is nothing to prevent the IAM files from being updated by other jobs or online systems.

Batch jobs that only read shared IAM files can be executed under IAM/RLS generally without any JCL or programming changes. Batch job steps that run relatively quickly, say in less than 15 to 30 minutes, and that update a relatively small portion of the records in the dataset, say less than 10% should also be able to use IAM/RLS recoverable files without any changes.
Batch jobs that update IAM/RLS files, which have long run times or update a large portion of the records in recoverable files may need to be changed, particularly if they are running concurrently with online systems or other batch jobs.

### Batch Syncpoint Processing

IAM/RLS has a syncpoint capability that can be used by batch applications. When recoverable files are being updated, the record locks are retained for records that are updated, added, or deleted until the job step terminates. If many records are being updated, then the number of locks held could be quite high, resulting in effectively locking out CICS transactions until the batch job completes. Such lock retention could also result in deadlocks between CICS and the batch job, resulting in failed requests. To prevent these types of problems, customers are encouraged to consider the use of the IAM batch syncpoint for such programs. Customers with other VSAM sharing packages may already have such logic built into their batch programs, and will only need to change the syncpoint program they are calling.

The IAM batch syncpoint service will perform a few different functions. First, it will make sure that all updated blocks for IAM files that are being processed by IAM/RLS for this job have been physically written to DASD. If blocks have not yet been written, such output I/O will be performed. Upon completion of any outstanding buffer writes, the batch syncpoint will then write out a syncpoint record in the journal that can be used if a back out recovery is necessary. Then, any record locks held by that batch job will be released.

### Implementing Batch Syncpoints

The amount of effort to implement a syncpoint within a batch job, along with the associated restart processing may be minimal, or may be rather substantial. This all depends on the nature of the updates being performed, the restart capabilities that the application program may already have, and how other updated files are handled. There are certain types of updates for which an application program can simply be restarted, and it can reapply those updates without the need for any recovery, and retain full data integrity. An example of this could be a job that updates addresses of customers. Such updates may not matter if they were already done before, the program will just end up repeating the update with the same data, or the application may have a way to check to see if the record was previously updated, and if so it would not have to reapply the update. For these types of programs, all that would need to be done is to periodically call the IAM batch syncpoint to release the locks of records updated to that point in time. For other applications, such as those that update account balances, a more complex approach may be necessary. These types of programs will need to implement a more complex syncpoint and restart mechanism, such that the program can be restarted from a specified point in time, and be able to resume processing from that syncpoint. When an application is able to do that, it can issue IAM batch syncpoints. If it needs to be restarted, back outs can be performed to for example the most recent syncpoint and the job will not perform any double updates. The batch application program will in effect, have to perform a processing checkpoint, and include in that the execution of the IAM batch syncpoint. The application will have to implement their own type of checkpoint processing, as MVS Checkpoint cannot be used with IAM files being processed under IAM/RLS.

Calling the IAM Batch syncpoint is easy. Just issue a call to IAMSYNC. No parameters are necessary. The program IAMSYNC is a small stub, and designed to be link-edited in with the application program. An alternative is for the application to LOAD the IAMSYNC module, saving the address and calling it as necessary.
As an alternative to revising the actual program, IAM/RLS has an automatic syncpoint capability. This ability is enabled by use of the IAM ACCESS override of AUTOSYNCPOINT=nnnnn. This override is specified on the job step that is updating IAM files through IAM/RLS for a selected key file that is processed using IAM/RLS. The AUTOSYNCPOINT specifies that after the specified number of I/O requests that result in record lock being acquired are performed, IAM will automatically invoke the batch syncpoint process. This facility is useful for circumstances where it is not possible to change the application programs.
20.70 IAM/RLS Data Recovery

Recovery Overview

One of the most important considerations when implementing IAM/RLS, or any type of data sharing, is how to recover should errors, media failures, or outages occur. The journaling and recovery tools provided with IAM will assist in the recovery of data for the IAM files that are processed under IAM/RLS. The intent of the information presented in this section is to assist you in planning for your data recovery needs, and to help insure that appropriate procedures are put in place for the protection of your data.

This section will address performing backout recoveries for failing batch jobs, and forward recoveries from restored IAM datasets. CICS provides the recovery for failed transactions with dynamic transaction backouts so IAM/RLS does not provide that capability.

IAM/RLS Journals

In order to perform recoveries with the IAM tools, you must provide for IAM/RLS Journaling, as described in Section 20.30. You’ll need to decide on what type of recoveries you’ll want to be able to perform. If you want to be able to backout updates from failing job steps, you’ll need to specify the journaling of before images. If you want to be able to recover files after restoring them from a backup, you’ll need to be collecting the after images. To provide the best possible recoverability, Innovation recommends that you journal both before and after images. This way, you’ll be able to perform whatever recoveries may be necessary. To insure that journaling is being performed for all of your files, it is recommended that you provide the appropriate ACCESS overrides to the IAM/RLS address space that will specify the level of journaling you need. For example, to specify journaling for all datasets, you can use the following IAM override control card for the IAM/RLS proc:

```
ACCESS DD=&ALLDD,JRNAD=BOTH
```

If you have other override cards, be sure to specify the JRNAD=BOTH on those as well, as the DD=&ALLDD override card is effective only for datasets that are not otherwise explicitly specified on override control cards.

Managing the IAM/RLS Journals

When allocating the journal datasets for IAM/RLS, it is recommended that you provide space for at least 24 hours of processing. Providing that amount of journal space across four to six journal datasets will provide the capability for performing backout recoveries within a 24-hour period without having to research what journal datasets need to be used. Innovation also recommends the use of the provided program, IAMJUTIL, with the REUSE option to copy the data from the active journal files to an accumulation dataset. This will keep the journal data within each IAM journal dataset until it is needed for reuse. Also, be sure to provide for an automated mechanism to offload journal data from the IAM/RLS journal datasets, to insure that there it will be ready when needed. This can be accomplished with the IAM/RLS JRNPROC= parameter, or by using automated operations software that looks for the message that an IAM journal file has been filled, and is being switched to the next journal file.
If you have sufficient DASD space, one method to handle the accumulation of the journal data is to have the procedure that offloads data from an IAM/RLS journal to place that data in a staging data set. If you have adequately sized journals, it will only be necessary to copy those journal records necessary for a forward recovery (i.e. the after images). Each time an IAM/RLS journal is filled, the data from the most recently filled journal is added to the end of the staging dataset. Then, on a scheduled periodic basis, such as daily, copy the data from the staging dataset, and put it out to a GDG on tape, then clear out the staging dataset. An example of what your offload proc might look like is shown below:

```
//JRNOFFLD PROC LOG=NULLFILE
******************************************************************************
/*
  /* COPY AFTER IMAGES FROM AN IAM/RLS JOURNAL
  /* TO AN ACCUMULATION DATA SET.
  /* MARK JOURNAL AS ELGIBLE FOR REUSE
  /*
******************************************************************************
/****** EXEC PGM=IAMJUTIL,REGION=6M
//SYSUDUMP DD SYSOUT=* 
//SYSPRINT DD SYSOUT=*  
//IAMJRNL DD DISP=SHR,DSN=&LOG    
//JRNLOUT DD DISP=MOD,DSN=PROD.IAM/RLS.JOURNAL.ACCUM  
//SYSOUT DD SYSOUT=* 
//SYSIN DD DISP=SHR,DSN=IAMSYS.RLS.PARMLIB(OFFLOAD)

Contents of IAMSYS.RLS.PARMLIB(OFFLOAD):

COPY AFTER,REUSE
```

Figure 20-13: Example of IAM/RLS Journal Offload to Accumulation Data Set (EX2070A)
The job to copy the data from the accumulation file to tape can use either IAMJUTIL or IEBGENER. If you wanted to keep the before images in the accumulation dataset, and only copy the after images to tape, you would want to use IAMJUTIL. An example is shown below:

```plaintext
//******************************************************
//*/  COPY AFTER IMAGES FROM AN IAM/RLS JOURNAL
//*/  ACCUMULATION DATASET TO A GDG ON TAPE.
//*/  THEN EMPTY THE ACUMMULATION DATASET.
//*/
//******************************************************
//COPYACUM EXEC PGM=IAMJUTIL,REGION=6M
//SYSUDUMP DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//IAMJRNAL DD DISP=OLD,DSN=PROD.IAM/RLS.JOURNAL.ACCUM 
//JRNLOUT DD DISP=(NEW,CATLG),DSN=PROD.IAM/RLS.DATA(+1), UNIT=TAPE 
//SYSOUT DD SYSOUT=*  
//SYSIN DD *  
COPY AFTER
/*
//MPTYACUM EXEC PGM=IEBGENER
//SYSUT2 DD DISP=OLD,DSN=PROD.IAM/RLS.JOURNAL.ACCUM 
//SYSUT1 DD DUMMY,DCB=PROD.IAM/RLS.JOURNAL.ACCUM 
//SYSPRINT DD DUMMY 
//SYSPRINT DD DUMMY 
```

*Figure 20-14: Example of copying journal data to GDG on tape (EX2070B)*

**Dynamic Job Backout**
IAM/RLS offers an optional capability to dynamically backout changes made to IAM/RLS files that were made by an abending job step. The use of Dynamic Job Backout is controlled by the IAM/RLS parameter DJB, as discussed in Section 20.10. When a job step that has updated IAM files managed by IAM/RLS abends, if Dynamic Job Backout can be used by that job, then IAM/RLS will internally schedule the backout. The backout will be performed under the IAM/RLS address space, with appropriate messages issued upon completion. Upon completion of the backout, IAM/RLS will release any record locks retained by the failing job step. If the abending job step had taken any IAM batch syncpoints, then the backout will be performed to the last syncpoint taken by the job step. If no syncpoints are found, then all updates made to IAM files managed by IAM/RLS will be backed out.
If you have not enabled Dynamic Job Backout, and a batch job step abends, you can use IAMJREST to perform a backout of the failing job step. You would not normally want to perform a backout of the updates made by the preceding job steps because those record locks would have been released upon normal step termination, and those records could have subsequently been updated by another batch job or online system. An example of a manual job backout is shown below. IAMJREST will automatically release any retained locks upon completion of the backout.

```
//BACKOUT EXEC PGM=IAMJREST,REGION=0M
//SYSPRINT DD   SYSOUT=*  
//SYSOUT     DD   SYSOUT=*  
//IAMINFO    DD   SYSOUT=*  
//SORTWK01   DD   UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK02   DD   UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK03   DD   UNIT=SYSDA,SPACE=(CYL,(50,10))
//SYSINDD*   
  RESTORE BACKOUT,JOBNAME=jobname,STEPNAME=stepname, 
  JOBID=JOBnnnnn
/*
```

Figure 20-15: Example of JCL for manual backout (EX2070C)
A forward recovery may be necessary if an IAM file is damaged due to hardware media failures, or due to software failures. The first step for such a recovery is to restore the IAM dataset from a backup copy. The backup could have been done by an FDRREORG, IDCAMS, or some other application, or it could have been done by a DASD management utility such as FDRABR or DFSMSdss. Once the dataset is restored, you can use IAMJUTIL to reapply all of the updates that had occurred to the dataset prior to the failure. For the forward recovery, you would want to specify the FROMDATE and FROMTIME being the time that the backup started, and specify the IAM dataset that is being recovered.

In the example below, there is the presumption that the current IAM/RLS journal datasets have all of the necessary data to perform the recovery, so no journal datasets are specified in the JCL.

```
//RECOVER EXEC PGM=IAMJREST,REGION=0M
//SYSPRINT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//IAMINFO DD SYSOUT=*  
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50,10)) 
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50,10))  
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50,10))  
//SYSIN DD *  
RESTORE FORWARD,DSN=bad.iam.file,FROMDATE=yyyyddd, 
FROMTIME=hhmmss  
/*
```

Figure 20-16: Example of a Forward Recovery (EX2070D)
25 USING IAM/PLEX RECORD LEVEL SHARING

IAM/PLEX Record Level Sharing

Overview

IAM/PLEX Record Level Sharing (RLS) is an optional additional cost feature of the IAM Version 9.2 product. With IAM/PLEX batch jobs, TSO users, and CICS regions concurrently running on multiple LPARs within a SYSPLEX can safely update shared IAM files. An IAM/PLEX is activated by defining an IAM/PLEX address space with a matching RLSGROUP on each LPAR that is participating with the IAM file sharing. The goal of IAM/PLEX is to provide performance along with ease of use in a multiple LPAR data sharing environment. IAM/PLEX is different in design and implementation than VSAM/RLS. In an IAM/PLEX only one IAM/PLEX instance will perform the actual I/O processing for any specific IAM file. The requests are passed to the target IAM/PLEX instance using SYSPLEX XCF communication. This eliminates the need for locking and caching structures in the Coupling Facility, and potential for multiple interactions with the CF for one logical I/O request.

Base requirements for an IAM/PLEX include z/OS, z/Series processor(s) and a coupling facility. For optimal performance the use of Channel-To-Channel (CTC) type connections are recommended for XCF communication.

The intention of the design of IAM/PLEX is to provide high availability of data access. IAM/PLEX does not guarantee continuous availability or eliminate single points of failure. As described below, IAM/PLEX in Version 9.2 does have some facilities that improve the recovery capabilities should a failure occur.

Version 9.2 Enhancements

Summary of changes for Version 9.2:

- Support for VSAM/RLS only I/O requests that may be issued from CICS application programs, including use of the operands Consistent, Repeatable, Nosuspend and Uncommitted. IAM/RLS is not currently able to provide support for the update browse operations due to the unique nature of the undocumented VSAM/RLS interfaces.
- Recognize and provide proper record locking function when the CICS parameter CILOCK=YES is being used with non-recoverable files.
- Special startup process that will activate the IAM VSAM Interface (VIF) and load the IAM/RLS and IAM/PLEX dataset name table without actually activating the IAM/PLEX address space. This is done by issuing the START command for the IAM/PLEX proc with PARM=DSNTLOAD. The IAM VSAM interface will be initialized and the dataset name table for IAM/PLEX will also be loaded into virtual storage. This will enable IAM to process data sets when the RLS Global Option indicates TABLE and REQUIRED before the IAM/PLEX address space is running.

IAM/PLEX Capabilities

IAM/PLEX provides the following capabilities:

- Ability to share IAM files for update with data integrity between concurrently executing batch jobs, TSO users, and CICS regions across multiple LPARs in a SYSPLEX.
- Locking during update processing is performed at the record level, utilizing the IAM/PLEX record locking function. Record locks can be reestablished after a
failure or shutdown of an IAM/PLEX instance to facilitate recovery processing.

- For recoverable files, record locks are held until the job or transaction terminates, or the job or transaction issues a SYNCPOINT.

- Potential deadlocks within the resources managed by each individual IAM/PLEX address space are detected, and the associated request will be failed to prevent a deadly embrace.

- Journaling and recovery capabilities are provided for data sets processed under IAM/PLEX and optionally for IAM datasets not processed under IAM/PLEX. Journaling in the IAM/PLEX RLS environment requires using the z/OS System Logger. Ability to dynamically back out updates made to files processed by IAM/PLEX when the batch job step that performed those updates abends.

- A callable batch syncpoint process is provided to prevent batch programs from locking out access to large portions of recoverable files when mass updates are being performed.

- An AUTOSYNCPPOINT feature that will automatically issue syncpoints without changes to the application can be used to avoid changing batch application programs when syncpoints are required.

- An Automatic Disconnect and Reconnect capability within CICS is provided to dynamically disconnect and reconnect to an IAM/PLEX instance that has terminated and subsequently has been restarted. Any pending in-flight updates by CICS transactions to IAM files owned by that IAM/PLEX instance at the time of its termination will be scheduled for backout. However since the IAM files are not available, any backout to an unavailable IAM file will fail. CICS then marks such failed backouts as pending Shunted Unit-of-Work (UOWs). Subsequently, when the terminated IAM/PLEX instance becomes available again, IAM will cause the shunted UOWs to be retried. Then IAM sets the files within CICS as being open and available for normal processing as IAM will have then fully reconnected to the restarted IAM/PLEX instance. Additionally, the CICS region will not have to be restarted or recycled as in prior IAM releases.

- An ISPF interface to monitor activity within the IAM/PLEX address space, and issue some of the available operator commands.

- Depending on the RLS selection criteria implemented at an installation, there generally will be little or no JCL changes required for IAM/PLEX. For CICS, the installation and activation of certain exits is required, as documented in the IAM/PLEX CICS Considerations in Section 25.50. For batch jobs that perform a large volume of updates to recoverable files being processed by IAM/PLEX, the implementation of the IAM Batch Syncpoint either with the AUTOSYNCPPOINT feature or through program changes and appropriate restart capabilities is necessary. Recoverable files are those for which before image journaling is being performed.
IAM/PLEX Definitions

To help explain how IAM/PLEX works and how it is used we'll start with some definitions of the terminology that is used when discussing IAM/PLEX. The terms are defined in more of a logical sequence than an alphabetic sequence.

RLSID: This is a unique 4-character identifier used to name each IAM/PLEX and IAM/RLS address space that is active within a SYSPLEX. It is specified in the initialization parameters for each IAM/PLEX and IAM/RLS address space via the RLSID= parameter, e.g. RLSID=RLS1. Each dataset that will be processed by an IAM/PLEX or IAM/RLS address space is assigned to an RLSID through the IAM/PLEX dataset name table, IAM overrides or defaulted to from the IAM Global Options Table to indicate which IAM/PLEX or IAM/RLS address space that will perform all of the I/O for that dataset.

IAM/PLEX Instance: Refers to an IAM/PLEX address space. Each IAM/PLEX instance is assigned a 4-character identifier called the RLSID as described above. Any particular application address space (i.e., batch job, CICS region) can directly connect to only one IAM/PLEX instance, through which it can access IAM files managed by other IAM/PLEX instances that are members of the same RLSGROUP. Each IAM/PLEX instance by default can perform both ROUTER and TARGET functions.

RLSGROUP: This refers to a group of related IAM/PLEX instances that form an XCF group, and therefore are able to communicate with each other. Any particular application address space (CICS region, batch job) can be connected to only one RLSGROUP, and can only access those IAM datasets that are being handled by an IAM/PLEX instance within that RLSGROUP. If no RLSGROUP is specified the IAM/PLEX instance effectively runs as an IAM/RLS instance.

ROUTER: Refers to the IAM/PLEX instance that an application (i.e., batch job or CICS address space) is directly connected to that is being used to send (route) I/O requests for a particular file to another IAM/PLEX instance that typically would reside on a different LPAR within the SYSPLEX. For any specific RLSGROUP there will be only one ROUTER IAM/PLEX instance per LPAR.

TARGET: Refers to the IAM/PLEX instance that is processing the I/O requests and handling record locking for a particular set of IAM datasets. The TARGET receives the requests from an application address space through a ROUTER IAM/PLEX instance. The TARGET IAM/PLEX instance is typically on a different LPAR than the ROUTER IAM/PLEX instance.

REMOTERLS: This attribute refers to an IAM/PLEX instance that performs file I/O only, it does not have any direct connections with application address spaces. It is an IAM/PLEX instance that can operate only as a TARGET. This attribute is given to IAM/PLEX instances that are running on the same LPAR as another IAM/PLEX address space that is a member of the same RLSGROUP.
IAM/Plex instances will connect to other IAM/Plex instances that are members of the same RLGROUP using SYSplex XCF services. Each IAM/Plex instance within a SYSplex is uniquely identified by a 4-character identifier called the RLSID. On any particular LPAR there can be only one IAM/Plex instance that can have ROUTER capabilities per RLGROUP. You can start more IAM/Plex instances for the same RLGROUP on the same LPAR using the REMOTERLS parameter during startup. If there is already a ROUTER IAM/Plex region for the RLGROUP on the LPAR, then the new one being started will default to REMOTERLS. The REMOTERLS address spaces will serve as TARGET-only IAM/Plex address spaces in the RLGROUP. Each IAM/Plex instance continually monitors through the z/OS XCF facility the status of the other IAM/Plex instances that are members of the same RLGROUP. Each of the IAM/Plex instances establishes a direct logical XCF connection with all of the other IAM/Plex address spaces that are members of the same RLGROUP.

An application address space (i.e. batch job step or CICS region) will create a connection with an IAM/Plex instance when the first IAM dataset that requires IAM/Plex processing is opened. The connection is made to the IAM/Plex instance with routing capabilities that is a member of the RLGROUP containing the IAM/Plex instance that is required to process the dataset being opened, which is the one with the matching RLSID. All IAM files that are subsequently opened by that application must belong to an RLSID that is being handled by this RLGROUP. IAM files will be opened only in the IAM/Plex instance that has the RLSID that was specified when the file was defined, loaded, via override, or entered in the DATASET NAME TABLE. The open request and any subsequent I/O requests will be sent to the TARGET IAM/Plex instance on whatever LPAR it resides in the SYSplex. On the next page is an example of an IAM/Plex configuration.
Below is a diagram of a basic 2-LPAR PLEX running one IAM/PLEX instance on each LPAR.

![Diagram of Basic IAM/PLEX Configuration](image)

In the above diagram there is one RLSGROUP named RLSPROD, which has two IAM/PLEX instances named RLS1 and RLS2. Each instance is running on a separate LPAR. For example let's say that CICS-1 wants to read a record for update in IAM DSN=PROD.B which is owned by IAM/PLEX_RLS2. CICS-1 issues a standard read for update, which IAM processes and sends the request from IAM/PLEX_RLS1 to IAM/PLEX_RLS2 using XCF services. IAM/PLEX_RLS2 will obtain the record lock, retrieve the record, then send the record through XCF back to the CICS-1 address space through IAM/PLEX_RLS1. In this scenario, IAM/PLEX_RLS1 is known as the router, and IAM/PLEX_RLS2 is the target.
Dataset Name Table

The Dataset Name Table provides the capability to group datasets by RLSID. A single Dataset Name Table must be used by all IAM/PLEX instances running in the SYSPLEX. The first IAM/PLEX instance that is started on an LPAR will read the dataset name table dataset and build the table in storage. All subsequent IAM/PLEX instances on the same LPAR will use the table already in storage, even the IAM/RLS address spaces not running in IAM/PLEX mode. If a change is required to the table, the CHANGEDSNT command can be used which will signal all the IAM/PLEX instances in the group to refresh their copy of the table. If an IAM/RLS address space running in single system or non IAM/PLEX mode requires the new changes to the table then the CHANGEDSNT command will have to be entered specifically for that IAM/RLS.

Record Locking

IAM utilizes its own record level lock manager. Records are locked by their key within each IAM data set being processed by IAM/PLEX. The record locks are maintained in 64-bit virtual storage in the IAM/PLEX address space that is responsible for the dataset containing the record. The IAM lock manager does check for potential deadlocks within the scope of the IAM data sets that it is managing. If waiting on the record lock for the current request will result in a deadlock, the request will be failed with a logical error. The trigger for IAM to release a record lock will depend on the environment in which the lock was requested.

For CICS transactions, records are locked by transaction and unit of work (UOW) identification. IAM determines from CICS which IAM files are recoverable, and will hold the record lock(s) until either a SYNCPPOINT is executed, or the transaction ends. If the IAM file is defined with the explicit option of JRNAD=NONE or the IDCAMS parameter LOG(NONE), then IAM will assume that the file is not recoverable under CICS. When an IAM file is detected as being not recoverable under CICS, then the record lock is only held from the time of the GET for UPDATE until the record is actually updated or erased, or for records being added, only for the duration of the actual add processing.

For other than CICS processing, that is batch jobs or TSO users, IAM will generally only hold the record lock(s) from the time of the GET for UPDATE until the record is actually updated or erased, or for records being added, only for the duration of the actual add processing. The exception to this is if IAM/PLEX is journaling before images of records, implying that a back out could be performed if there is a failure, then the record lock(s) will be held until the program either calls the IAM batch synpoint, an AUTOSYNCPPOINT is taken, or the job step terminates. If a job step abends while processing a recoverable file, then any record locks obtained for that job step will be retained until a recovery takes place. If there were any other jobs or CICS transactions waiting for the record locks that are being retained until recovery, those requests will be failed when the abend has been detected. The recovery can be performed by IAM’s Dynamic Job Back Out function, by IAMJREST, or by whatever other procedures or recovery software that may be available. Information about any retained locks can be found by using the IAM/PLEX operator command DISPLAY,RETAI NEDLOCKS, described in Section 25.40. Such retained locks will be released upon successful recovery by IAMJREST or Dynamic Job Back Out. The retained locks can also be released by the RELEASELOCKS command.

Record Lock Recovery

IAM/PLEX has an optional capability to reestablish record locks on restart of an IAM/PLEX address space after either an unplanned or a planned outage. Only the record locks that are for recoverable IAM datasets will be reestablished. The record lock status for locks eligible for recovery are kept in a separate lock recovery dataset. To minimize the performance impact, the records will only be periodically written to disk after a specified lock check point time interval.
When the IAM/PLEX address space is started, the record lock recovery process will read the lock recovery dataset to reestablish record locks. If the last status was not from a proper shut down then the IAM/PLEX journals will be read from the time of the last lock checkpoint to update the record locks from the recorded activity. Once the record locks are reestablished, then the IAM/PLEX address space is ready to resume processing.

Journaling

In addition to handling the I/O and locking services, the IAM/PLEX address space can also be used for journaling before and / or after images of updated records. When using journaling with IAM/PLEX the System Logger must be used and the LOGSTREAM must be defined to a structure in the coupling facility. The IAM journaling services are primarily provided to allow for the back out of updates performed by failing batch job(s), for record lock recovery, or to perform a forward recovery of updates if a file has encountered media damage. CICS will handle its own transaction back out and other recovery as it does today, using its own logging mechanisms that is independent of the journaling provided by IAM. Therefore unless either record lock recovery is active or journaling of CICS transactions is explicitly requested, the before images for files processed under CICS will not be written to the IAM/PLEX journal. To request IAM/PLEX to journal the before images from CICS transactions users can specify the IAM/PLEX startup parameter CICSJOURNAL=YES.

Information and instructions on setting up IAM/PLEX journaling is presented in Section 25.30.

Replication Logging

Support has been added to produce journal records to a System Logger logstream in the same format written by CICS in the CICS journals to support the use of replication software such as the Record Recovery Data Facility RRDF™ from E-Net. This logging facility is activated by adding the ALTSYSLG DD statement to the IAM/RLS or IAM/PLEX proc. The DD statement should specify the name of the alternate logstream to be used for these records. This is a 16 character name like, IAMLOGR.ALTGRP01. Then you have to specify the IAM ACCESS override ALTLOGGER for the datasets that you want these alternate journal records written for. The ACCESS override would have to be in the overrides used by the IAM/RLS or IAM/PLEX address space, and System Logger must also be active for the normal IAM/RLS or IAM/PLEX journals. Only after images will be written to this log which are adds, updates, and deletes.

Dynamic Job Back Out

An optional Dynamic Job Back Out function for IAM files opened through IAM/PLEX is available. Whenever a job step abends, all updates done by that job step can be automatically removed by Dynamic Job Backout. Control of Dynamic Job Back Out is provided through IAM/PLEX DJB startup parameter. If permitted by the startup parameter, DJB can also be controlled via an IAM ACCESS Override. If the batch job step or TSO user has taken IAM syncpoints, then the backout will be performed to the most recent syncpoint taken by the job step prior to abending. If no IAM syncpoints have been taken, then all of the updates performed by the failing job step will be removed on files accessed through IAM/PLEX. Upon successful completion of the Dynamic Job Backout, all of the retained locks for the failing job will be released.
IAM/PLEX Record Level Sharing

Security

The IAM/PLEX address space will have to be given the appropriate security authority for the IAM files that are going to be processed by IAM/PLEX Record Level Sharing. This will include CREATE authority for files that can be extended to additional volumes. IAM does issue the RACROUTE macro within the individual job’s address space to validate that the requesting user does have authority to read or update as appropriate while the IAM data set is being opened prior to requesting that IAM/PLEX open the data set. If the RACROUTE indicates that the user or job does not have authority to access the data set, then the OPEN request is failed. The failing job will receive an IAMW18 error message. Additional security will be performed by z/OS services when IAM attempts to take an extent.

Reliability

IAM/PLEX utilizes the various z/OS error handling and recovery facilities to recover from errors and abend conditions that may occur. The two goals in providing error recovery routines are to provide continuous availability of the services being provided by the IAM/PLEX address space, and secondly to automatically collect enough information about any failures such that problem determination and correction can be performed from the single failure. The job that had submitted an I/O request to IAM/PLEX that has experienced a failure will generally be failed with a VSAM logical error code. It will be up to the program to decide whether it can continue processing.

The error data collection will include various messages to the job log, system log, and to the RLS log indicating the abend code, general registers, access registers if applicable, and the failing module. The error routines do attempt to not duplicate diagnostic information that is produced by z/OS, but rather provide additional diagnostic information to be combined with the information provided by z/OS. The error information contained in these messages may be sufficient for problem determination and resolution, particularly if a problem had been previously reported to Innovation. If the error occurred within the IAM/PLEX address space, an error trace table will be kept in storage for reference, particularly in situations where multiple errors have occurred. Most error situations will also result in a request for a dump to be taken to a system dump data set, which may include both the IAM/PLEX address space and possibly the address space of the job that submitted the failing request. The IAM index space associated with the IAM/PLEX address space may also be dumped.

Serviceability

To aid in serviceability a mechanism is provided for system support personnel to apply critical fixes to the IAM modules within the IAM/PLEX address space, without the need to shut down the address space. This facility can also be used to backout fixes should they cause other problems. Refer to the IAM/PLEX operator commands APPLY and RESTORE in Section 25.40.

HiperDispatch Enhancements

Some performance issues were discovered when running with HiperDispatch mode enabled on LPARs that had more than 4 engines assigned when doing heavy batch processing. The key problem areas were determined to be local lock contention, and dispatching overhead that was impacted by the lock contention. For most I/O requests, IAM used fewer cycles than it was taking to dispatch the work. Based on that information a number of enhancements have been made to the IAM/PLEX processing to alleviate those issues, which include the following:

- Implemented use of the Perform Locked Operation (PLO) instruction when updating various queues that had been serialized with the local lock.
- Revised key paths to eliminate use of various SVC services that could serialize on various z/OS locks including GETMAIN, FREEMAIN, and MODESET. To improve short term storage acquisition and release was revised to utilize internal
cell pool structures.

- Implemented use of z/OS high performance Pause and Release services that replaced WAIT and POST for scheduling the I/O tasks, and replaced SUSPEND and RESUME used to ready completed I/O requests for batch jobs.

- Revised I/O task dispatching and management, as explained below.

The IAM/PLEX I/O task management has been revised to provide improved throughput by using fewer I/O tasks. Rather than attaching a new task when an I/O request is received and there are no available I/O tasks, the I/O request is put on a single common work queue. Then the currently active I/O tasks will pull work from the work queue whenever they are done with the I/O request in progress, eliminating any z/OS services from being involved. I/O requests are added to and removed from the work queue with the PLO instruction, avoiding use of z/OS locks. When an active I/O task finds the work queue empty, it uses the Pause service to cause it to wait.

The IAM/PLEX workload manager will manage the number of active I/O tasks between the MINIOTASK and MAXIOTASK values to provide good throughput with the least number of I/O tasks. When an I/O task is no longer necessary to handle the workload, it is placed on an idle queue, where it will remain ready to be reactivated when necessary. When the number of I/O requests on the work queue exceeds internal thresholds, an I/O task will be reactivated if available. If none are available, then an additional I/O task will be attached as long as the total number of I/O tasks is less than the MAXIOTASK value. The values are reported on in the periodical RLSINFO reports if you have them enabled.

It is expected that these changes will reduce the number of I/O tasks being utilized. To help facilitate that, we encourage users to reduce the MINIOTASK value to 4. You can adjust MINIOTASK to determine what value provides the best throughput for your workload, however making it too high can be counter productive with HiperDispatch enabled.

Record lock contention can result in an increase in active I/O tasks. From the IAM/PLEX perspective, a wait on a record lock can be a long wait. When an I/O request has to wait for a record lock, IAM checks to see how many other tasks are waiting for locks. If this new request would result in all active tasks waiting for record locks, it will invoke the workload manager to add another task. If the IAM/PLEX region is already at MAXTASK IAM/PLEX will prevent a complete locked situation by failing I/O requests at that time that need to wait on a lock with a logical error code of x’18’. While record lock contention is generally low, this could be a factor in how high a value to use for MAXIOTASK.
25.03 IAM/PLEX Implementation

This section provides the outline of the procedure to follow to implement IAM/PLEX. Innovation recommends that you start out with a very limited implementation, and then as you become comfortable with IAM/PLEX, the use of it can be expanded to more datasets. Many of the parameters that control IAM/PLEX can be easily modified while IAM/PLEX is active, which will minimize the need to deactivate IAM/PLEX to broaden the implementation to more datasets. Within any CICS region or batch job, there can be some IAM datasets using IAM/PLEX, while other IAM datasets are not using the IAM/PLEX services.

Careful planning and consideration must be given to how to recover data when a software or hardware error occurs. This is particularly true if you've never used any type of data sharing software in the past. For example, if a batch job abends, you may not be able just to restore a dataset and restart the job because of the potential loss of data from CICS users concurrently updating the dataset. IAM does provide journaling and recovery capabilities that can be used to perform various types of recoveries. You should determine ahead of time what your needs are, which can be different for your different datasets and applications. Careful planning and testing of recovery procedures will be of great benefit to successfully implement IAM/PLEX. Refer to Section 25.70 for more information.

The various tasks necessary to implement IAM/PLEX are shown below. Please be sure to review all of the sections to make sure that nothing is missed as you develop your implementation plan.
<table>
<thead>
<tr>
<th>CHECK</th>
<th>STEP</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>Set up the IAM/PLEX proc within the appropriate system proclibs, and allocate any required parameter libraries. Refer to Section 25.05 for the details on the IAM/PLEX proc and address space requirements. A separate proc will need to be created for each IAM/PLEX address space that is to be executed. Information on allocating the optional record lock recovery dataset is in Section 25.06.</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Review and select the appropriate values for the various IAM/PLEX parameters. Review Section 25.10 for detailed information on the IAM/PLEX parameters.</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Set up the appropriate criteria for your installation for the IAM/PLEX automatic dataset selection. Refer to Section 25.20 for the details on automatic dataset selection.</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Update your coupling facility policies to define a STRUCTURE, and LOGSTREAM datasets that will use that STRUCTURE in the coupling facility. Each RLSGROUP, if you use more than one, must have its own LOGSTREAM. Up to 10 LOGSTREAMS can be defined using the same STRUCTURE in the coupling facility. Refer to Section 25.30 for information on IAM/PLEX journaling.</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Review your COUPLEExx member in parmlib and make sure that there are CLASSDEF and PATHIN and PATHOUT statements coded. This sets up defaults for the transport classes used by XCF for signaling purposes. Please refer to MVS Setting Up a Sysplex manual for details on setting up the COUPLEExx member of parmlib, and to section 5.0 in that manual for Planning Signaling Services in a Sysplex.</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Review Section 25.50 on CICS Transaction Server considerations. Install and activate the necessary CICS exits provided by IAM in every region that will be using IAM/PLEX. The CICS exits are critical for IAM to perform record locking for CICS. Failure to properly install and implement them will result in record lockouts and other failures. There are no application or JCL changes required for CICS to use IAM/PLEX.</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>Review Section 25.60 on batch application considerations. For batch jobs that update a large portion of the records in a recoverable file that is processed by IAM/PLEX use of the IAM/PLEX Batch Syncpoint program or AUTOSYNCPPOINT is required, along with providing for recovery and restart should such programs experience software or hardware failures.</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>Plan and develop procedures for data recovery, to be prepared in case of unexpected software or hardware errors occur. Make sure that your implementation for IAM/PLEX journaling will provide the necessary data to perform data recoveries that will be necessary when failures occur. Refer to Section 25.70 for information on recovery of data for IAM files being processed under IAM/PLEX.</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>Review Section 25.40 on the IAM/PLEX operator commands, for information on the various operational requests that can be made of IAM/PLEX.</td>
</tr>
</tbody>
</table>
25.05 IAM/PLEX JCL Procedure

IAM/PLEX runs as a started task in a z/OS system. The IAM/PLEX procedure runs continuously as a non-swappable address space, providing services to the various jobs and CICS regions that are sharing IAM files. In general, IAM/PLEX will be a task that should be kept running all the time. However, IAM/PLEX can be shut down and restarted whenever it may be necessary to do so without impacting processing for IAM datasets that are not using IAM/PLEX.

Because IAM/PLEX is an I/O service provider, you will need to make sure that the IAM/PLEX address space is provided with a high enough priority to service the CICS transactions and batch jobs to meet their response time requirements. It probably should be given a priority equal to or higher than the most critical CICS region or batch job to insure that adequate service is provided. Specifying too low of a priority will result in poor response times for CICS transactions and elongated run times for batch jobs. If you are using the Workload Manager (WLM) to manage the workload in your SYSPLEX then you should consider assigning the same service class to IAM/PLEX that is used for the DB2 Lock Manager.

If you are installing IAM/PLEX, you can replace the execution of the IAMSTART procedure with the start up of IAM/PLEX. IAM/PLEX will automatically run IAMSTART to install the IAM VSAM Interface, if it is not already active. Note that subsequently shutting down IAM/PLEX will not deactivate the IAM VSAM interface, the IAM VSAM interface will be left active.

Users can run one full function IAM/PLEX instance per LPAR of the same RLSGROUP. You can have other IAM/PLEX instances of the same RLSGROUP on the same LPAR. If the REMOTERLS parameter is not specified, then it will default to a REMOTERLS if there is already a ROUTER IAM/PLEX instance for the RLSGROUP on the LPAR. A REMOTERLS region is a TARGET-only region and cannot be connected to directly by a batch job or CICS region. Users can start additional IAM/PLEX regions that are part of a different RLGROUP on the same LPAR. Users can also have a mix of IAM/PLEX regions and IAM/RLS regions on the same LPAR. There could be several reasons to run multiple IAM/PLEX address spaces. For example, one might want to isolate different applications such that a failure with one application grouping will not impact other unrelated applications. Another reason might be for billing purposes, to be able to isolate as much as possible the processing for different applications or customers by eliminating common server address spaces. Or, it could be necessary or desirable for security purposes to have certain applications isolated from others.

Whatever the reason customers that desire to utilize multiple IAM/PLEX address spaces will need to have well defined groups of datasets and processing that can be isolated from each other to make this possible.
The use of multiple IAM/PLEX address spaces is subject to the following restrictions:

1. A batch job, CICS region, or TSO user can only be directly connected to ONE IAM/PLEX instance. All IAM datasets that require IAM/PLEX processing for that user address space MUST be handled by an IAM/PLEX instance that is a member of the same RLSGROUP as the IAM/PLEX it is connected to.

2. An IAM dataset can only be opened by ONE IAM/PLEX instance at a time.

3. Each IAM/PLEX instance must be started with its own unique proc and name.

4. Each IAM/PLEX instance in an RLSGROUP must use the SYSTEM LOGGER, and the LOGSTREAM must be defined as part of a STRUCTURE in the coupling facility.

5. All IAM/PLEX instances in the SYSPLEX must use the same Dataset Name Table.

6. Each IAM/PLEX instance requires its own system LX. A system LX is the facility through which cross memory access to the IAM/PLEX address space is authorized. Due to system limitations on the number of system LX’s, this will restrict the maximum number of IAM/PLEX address spaces that can be used unless the address space is started with the REUSASID=YES parameter on the START command for the IAMRLS procedure. If you are going to use the REUSASID=YES parameter, make sure that the DIAGxx member in parmlib also specifies REUSASID(YES).

IAM/PLEX PROC

In the ICL (Installation Control Library) that was part of the install package, is an example of the IAM/PLEX procedure, which is shown below. This example can be edited and copied into one of your system proclibs. You will need to change the dataset name of the IAM load library, and the dataset name of the card image PDS to be used for parameters and overrides. You may also desire to change the default member names for the IAM/PLEX parameters and for the IAM/PLEX override.

```
//IAMPLEX  PROC MBR=PLXPRM1,OVR=PLXOVRID,TBL=PLXDSNTB
//IEFPROC  EXEC PGM=IAMPLEX,MEMLIMIT=8G,
    //REGION=0M,DPRTY=(15,15),TIME=1440
//STEPLIB  DD   DSN=your.iam.loadlib,DISP=SHR
//SYSLIB   DD   DSN=your.iam.loadlib,DISP=SHR
//SYSPRINT DD   SYSOUT=X
//RLSLOGDD DD   SYSOUT=X
//SYSABEND DD   SYSOUT=X
//RLINFO   DD   SYSOUT=X
//RLKJRNLD DD   DISP=SHR,DSN=your.lock.recovery.dataset
//IPARMLIB DD   DISP=SHR,DSN=your.iam.icl(&MBR)
//IAMOVRID DD   DISP=SHR,DSN=your.iam.icl(&OVR)
//IAMDSNTB DD   DISP=SHR,DSN=your.iam.icl(&TBL)
```

Figure 25-2: Sample of IAM/PLEX PROC: Member IAMPLEX in ICL
The PARM='TEST,JOBNAME=(JOBX*)' parameter can be used in your IAM/PLEX startup procedure to start a test version of IAM's VSAM interface (VIF). You will have to set up a different PROC using a unique RL SID and start the PROC with a STEPLIB to the load library with the level being tested. To run a job using the test IAM/PLEX you use the same STEPLIB and specify the RL SID of the test IAM/PLEX. If you start the IAM VSAM Interface (VIF) before starting IAM/PLEX, then you will have to put the PARM='TEST,JOBNAME=(JOBX*)' in the procedure that you use to start VIF. An example of a test IAM/PLEX proc is shown below. The test job names are specified by the JOB parameter when starting the IAMRLST procedure.

```plaintext
//IAMPLEXT PROC MBR=RLSPARMT,OVR=RLSOVRID,TBL=DSNTB
//IEFPROC EXEC PGM=IAMPLEX,MEMLIMIT=4G,
//          PARM='TEST,JOBNAME=(JOBX*)',
//          REGION=0M,DPRTY=(15,15),TIME=1440
//STEPLIB DD DSN=your.iam.loadlib,DISP=SHR
//SYSLIB DD DSN=your.iam.loadlib,DISP=SHR
//SYSPRINT DD SYSOUT=X
//RLSLOGDD DD SYSOUT=X
//RLSINFO DD SYSOUT=X
//SYSABEND DD SYSOUT=X
//RLOKJRNL DD DISP=SHR,DSN=your.lock.recovery.dataset
//IPARMLIB DD DISP=SHR,DSN=your.iam.icl(&MBR)
//IAMOVRID DD DISP=SHR,DSN=your.iam.icl(&OVR)
//IAMDSNTB DD DISP=SHR,DSN=your.iam.icl(&TBL)
```

Figure 25-3: Sample of TEST IAM/PLEX PROC Member IAMPLEXT in ICL
IAM/PLEX DD STATEMENTS

The DD statements needed in the IAM/PLEX proc are described below.

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Specifies the IAM load library data set. This DD is optional if IAM is in the system link list.</td>
</tr>
<tr>
<td>SYSLIB</td>
<td>Required DD statement that specifies the IAM load library data set.</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>Required DD statement, specifying a sequential output data set that is normally printed. If specified as SYSOUT, make sure the class used is one that is not automatically purged, in case the output is needed for problem diagnosis.</td>
</tr>
<tr>
<td>RLSLOGDD</td>
<td>Optional DD statement, specifying a sequential output data set that is normally printed. This file contains a log of activity and error messages. If specified as a SYSOUT, make sure that the class used is one that is not automatically purged, in case the output is needed for problem diagnosis. If not included in the IAM/PLEX PROC, then IAM/PLEX will dynamically allocate a printed log file to SYSOUT, using the SYSOUT class value specified by the SYSOUT=x IAM/PLEX parameter. If the SYSOUT class is not specified, then the default SYSOUT class for the started proc will be used. By allowing this data set to be dynamically allocated, you can spin off the current log data at any time by using the IAM/PLEX LOGSWITCH command.</td>
</tr>
<tr>
<td>SYSABEND</td>
<td>An optional, but highly recommended DD statement that will be used when a dump is taken due to an abend condition. An alternative would be to use a SYSMDUMP DD statement. Use of SYSUDUMP is not recommended, because it will contain insufficient information for problem diagnosis.</td>
</tr>
<tr>
<td>RLSINFO</td>
<td>An optional DD statement that is required only if you want the IAM/PLEX stats reports to be generated as specified by the STATS parameter in the startup parameters.</td>
</tr>
<tr>
<td>RLOKJRNLD</td>
<td>An optional DD statement that is required to utilize the record lock recovery / persistence function. See Section 25.06 for detailed description, allocation and usage guidelines.</td>
</tr>
<tr>
<td>IPARMLIB</td>
<td>An optional DD statement, which specifies an 80-character card image input data set containing various parameters and options for IAM/PLEX. This data set is typically a PDS.</td>
</tr>
<tr>
<td>IAMOVRID</td>
<td>An optional DD statement that specifies an 80-character card image input data set, which contains the IAM override statements. This provides a mechanism for indicating parameters, such as number enabling journaling for all files processed by IAM/PLEX.</td>
</tr>
<tr>
<td>IAMDSNTBD</td>
<td>An optional DD statement that specifies an 80-character card image input data set. This data set contains the dataset selection and dataset exclusion lists for IAM/PLEX processing.</td>
</tr>
<tr>
<td>IAMINFO</td>
<td>An optional DD statement that will contain the standard IAMINFO reports produced whenever an IAM file is closed. As this will generate a significant amount of output, it is recommended that it not be specified and that users enable the IAM SMF records to be recorded which contain the same information.</td>
</tr>
<tr>
<td>ABNLIGNR</td>
<td>A DD DUMMY to bypass Abend-Aid dumps.</td>
</tr>
<tr>
<td>ABNLDUMP</td>
<td>A DD DUMMY to bypass Abend-Aid dumps.</td>
</tr>
</tbody>
</table>
IAM/PLEX Record Lock Recovery

Record lock recovery is an optional process that enables IAM/PLEX to reestablish the record locks for recoverable files when an IAM/PLEX instance is restarted after a shutdown or failure. This function enhances the data integrity by providing persistent record locking within the IAM/PLEX sphere, which will enable recovery of the work that was in process at the time of the incident prior to any other attempts to update those records through IAM/PLEX. Using a two phase recovery process, as explained below, IAM will re-establish the record locks as they were at the time of failure based on data from both the record lock journal and the IAM/PLEX System Logger journal.

Eligibility requirements for record lock recovery are that the locks being held are for a recoverable IAM data set, which is defined as being a data set that is using the IAM/PLEX journaling that includes before images. When record lock recovery is active, it will also automatically result in before images for CICS recoverable files to be written in the IAM/PLEX journals. The journal records are necessary for the recovery process.

The record lock recovery process retains information about the record locks that are currently held on disk in an IAM dataset that is referred to as the Record Lock Journal. The presence of the Record Lock Journal in the IAM/PLEX JCL is the indication that the record lock recovery process is to be activated. Each IAM/PLEX instance will need to have its own Record Lock Journal data set. To minimize the performance impact, the I/O on the data set utilizes deferred writes which delays the actual physical writing of the record lock status. At periodic intervals, referred to as the record lock checkpoints, all the buffers containing updated record lock data are written out to DASD and the current position in the IAM/PLEX journal is recorded.

The checkpoint interval can be set through the IAM/PLEX parameter LOCKCHKPT, which will specify the checkpoint interval in minutes. The default if the LOCKCHKPT parameter is not provided is 5 minutes. Using shorter times may increase the overhead during regular processing, but decrease the recovery time. Using longer checkpoint times will have the opposite effect.

The actual record lock recovery process is done in two phases while the IAM/PLEX instance is starting up. The record lock recovery must complete before the IAM/PLEX instance will start processing I/O requests. The first phase is to read through the Record Lock Journal dataset and establish the record locks based on the information from that dataset. If the status indicates that the IAM/PLEX instance had been shut down properly, meaning that the lock recovery journal has all of the lock data, then the recovery is complete.

If the IAM/PLEX instance was not normally shut down, then the record lock recovery process will go into the second phase. The second phase will read the IAM/PLEX system logger journal. Positioning is established at the place in the journal where the last record lock checkpoint had occurred, then proceed forward in the journal from that point. The recovery process will then update the record locks based on the activity indicated by the journal records, which includes updating the lock recovery journal.

It is recommended that users backup the lock recovery journal prior to restarting the IAM/PLEX instance from a failure. This can be done with a copy, a backup with FDRDSF, ABR or similar product, or by an IDCAMS REPRO. If a failure should occur on the restart of the IAM/PLEX instance, one will then be able to restore the lock recovery journal and retry the restart process again, otherwise a cold start may be necessary.
On completion of the record lock recovery, a summary of the activity performed by the lock recovery process will be printed on the RLSLOGDD dataset. Then the IAM/PLEX instance will finish initialization and start accepting requests. To find out what record locks have been recovered one can either use the IAM/PLEX display retained locks command or use the IAMBMON facility under the IAM ISPF panels. If any of the work is not going to be recovered, the associated locks can be released using the IAM/PLEX release locks command.

To activate the record lock recovery function, a lock recovery journal dataset must be pre-allocated using an IDCAMS DEFINE prior to use for each IAM/PLEX instance that record lock recovery is going to be used. Each instance will only have the record locks for the IAM datasets that it is servicing. Any IAM/PLEX instance that does not directly do I/O to any file will not need a lock recovery journal. The lock recovery journal dataset contains a control record, lock owner records, and record lock records. The dataset will be formatted when it is first used based on the IAM/PLEX parameters of MAXJOBS, MAXLOCKS, and MAXTRANS. Below is an example of the JCL and control cards that are used to define the lock recovery journal dataset:

```
//DEFRLKJR EXEC PGM=IDCAMS
//IAMPRINT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=* 
//IAMOVRID DD * 
CREATE DD=&ALLDD,DATACOMP=NO,DSNTYPE=LARGE,RELEASE=NO 
//SYSIN DD * 
DELETE (IAMV.RLOKJRNL.rlsid.CLUSTER) 
DELETE (IAMV.RLOKJRNL.rlsid.CLUSTER) NOSCRATCH 
SET MAXCC=0 
/* RECOMMENDED SPACE ALLOCATION VALUES: */ 
/* CYL(  60       6)  WITH MAXLOCKS UP TO   133,103  */ 
/* CYL( 120      12)  WITH MAXLOCKS UP TO   266,239  */ 
/* CYL( 230     23)  WITH MAXLOCKS UP TO   532,453  */ 
/* CYL( 460     46)  WITH MAXLOCKS UP TO 1,064,957 */ 
/* CYL( 900    90)  WITH MAXLOCKS UP TO 2,129,903 */ 
/* CYL(1350   135)  WITH MAXLOCKS UP TO 3,193,873 */ 
/* CYL(1800   180)  WITH MAXLOCKS UP TO 4,194,301 */ 
DEFINE CLUSTER - 
(NAME(IAMV.RLOKJRNL.rlsid.CLUSTER) - 
 CYL( 900  90) - 
 VOL(* * *) - 
 CISZ(12288) REUSE - 
 STORCLAS(TSTDATA) - 
 OWNER($IAM) - 
 RECORDSIZE(100 328) KEYS(5,0) - 
 FREESPACE(0,0) - 
 SHAREOPTIONS(2,3)) 
 LISTC ENTRIES(IAMV.RLOKJRNL.rlsid.CLUSTER) ALL 
/* 
```
When using the above JCL, users will need to change the dataset name and the STORCLAS. Also the space requested should be updated based on the MAXLOCKS value they are using. Please note that one needs to use the next larger value from the table if the MAXLOCKS value chosen exceeds the prior value. For example if MAXLOCKS=133104 is specified, then the space allocation used must be the one for up to 266,239 lock records. All of the other parameters and overrides shown in the example above must be used to provide proper functioning and performance of the lock recovery function. If the value for MAXLOCKS is ever increased, the lock recovery journal should be re-allocated with the required space values indicated.

After defining the lock recovery journal, users must then make the following changes to the IAM/PLEX JCL:

- Add the RLOKJRNL DD statement to the IAM/PLEX JCL specifying the newly defined dataset.
- Add a card in the IAMOVRID member input that specifies the following:
  ACCESS DD=RLOKJRNL,NORLS,JRNAD=NONE

When the IAM/PLEX instance is started, IAM/PLEX will automatically determine that the dataset is unloaded and format the dataset with the appropriate number of pre-formatted records. This pre-formatting is completed quickly and will provide performance benefits for the operation of the lock recovery function. If any of the key IAM/PLEX parameters are changed, that is MAXLOCKS, MAXJOBS, or MAXTRANS, then the dataset should be re-initialized by specifying the IAM/PLEX parameter of LOCKSTART=COLD which will trigger the reformating of the dataset. If the value for MAXLOCKS has been increased then the dataset may need to be deleted and re-defined with the appropriate amount of space.

It is highly recommended that a backup copy of the lock recovery journal be taken whenever the IAM/PLEX has come down, whether it was scheduled or unscheduled. The dataset can be backed up with a copy, FDRDSF, ABR or similar type of backup, or by an IDCAMS REPRO. Then if there is a problem on the restart, the lock recovery journal can be restored prior to retrying the restart.

Backups can be taken at other times while the IAM/PLEX address space is running, however the restored copies can not really be used to restart the IAM/PLEX for other than a cold start (LOCKSTART=COLD). This is because the restored copy is far from being synchronized with the actual state when the IAM/PLEX address space had terminated. A cold start is also likely to be required when starting IAM/PLEX address spaces at disaster recovery sites.

There generally should not be any need to reorganize the lock recovery dataset. The define parameters and overrides in the example above were carefully chosen along with the way the dataset is processed to minimize the need for use of the overflow area. This is particularly true when appropriate values were specified on the IAM/PLEX parameters such that they were sufficient to meet the processing that was done. One can either look at a LISTCAT output or at the IAMINFO report to determine if there are a lot of records in overflow. If there are, then one can check on the maximum locks used from the RLSINFO report that is produced or from the IAMBMON display to make sure that sufficient lock records had been formatted. One should also check that all of the appropriate parameters in the define example above were used. It may be necessary to delete and redefine the dataset such that the overflow area usage will be minimized. Innovation technical support can provide assistance if needed.
25.10 IAM/PLEX Parameters

There are three sources of parameters for controlling IAM/PLEX usage. These include the IAM/PLEX start up parameters, specified by a parmlib mechanism on the IPARMLIB DD statement, the IAM Overrides, and the IAM Global Options Table. The IAM/PLEX parmlib parameters are completely discussed in this section. There is also a discussion of relevant IAM Overrides with IAM/PLEX in this section, the full IAM Override documentation is provided in Section 30 of this manual. The main global option affecting IAM/PLEX has to do with specification of automatic eligibility for IAM/PLEX processing, which is discussed in Section 25.20.

IAM/PLEX PARMLIB

The parameters that can be specified for the IAM/PLEX address space are presented below. These parameters are passed to IAM/PLEX via a parmlib mechanism, which is specified via the IPARMLIB DD statement. If you use the example IAM/PLEX procedure, then the IAM/PLEX parmlib member is specified using the MBR= operand. Each IAM/PLEX instance must have its own parmlib member to specify a unique RLSID and the RLSGROUP that the IAM/PLEX instance belongs to.

The parmlib is read to establish the parameter settings during IAM/PLEX initialization. Additionally, users can issue the “CHANGEPARM” command, which will cause an active IAM/PLEX address space to reread the parmlib member, and cause an immediate update for many of the values. Any special considerations with changing the various parameter values will be discussed in the description of the individual parameters below.

The parmlib dataset must be an 80-column card image dataset, with parameters and values coded in columns 1 through 71. Each card image can contain a single parameter with its values. Multiple parameters can be specified on a single card, but must be separated by a comma and no spaces.

Comment cards can be included, and are indicated by an "*" in the first column of the card. Examples of parmlib are provided in the ICL (Installation Control Library). Customers may use the ICL for the parmlib data set, if so desired.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICSJOURNAL=</td>
<td>Specifies whether IAM should journal the before images for updates performed by CICS transactions. Valid values are YES or NO. The default value is NO, because CICS normally handles transaction backout and before image journaling itself. If the Record Lock Recovery is being used, then it will automatically default to YES.</td>
</tr>
</tbody>
</table>
IAM/PLEX Parameters

CICSLOCK=

Specifies the action that IAM/PLEX will take when record lock contention is encountered for a CICS transaction. Three sub-parameters can be provided. The first instructs IAM/PLEX what action to take when the owner of the lock is another CICS transaction. The second instructs IAM/PLEX what action to take when the owner of the lock is other than a CICS transaction, such as a batch job or TSO user. The third tells IAM/PLEX that if it is going to fail the request due to lock contention, it is just to pass a logical error code back to the caller, or abend the transaction.

One of the following keywords can be provided to indicate what action IAM/PLEX will take when the owner of the record lock is a CICS transaction:

- **CICSWAIT** specifies that the requestor of the lock will wait until the record lock is available when the owner is a CICS transaction.
- **CICSTIMEOUT** specifies that the requestor of the lock will wait for a short time as specified by the CICSTIMEOUT parameter (see below), if the owner of the lock is a CICS transaction. If the lock is still not available, then the request will fail.
- **CICSNOWAIT** specifies that the request for the record lock will be failed if the owner is a CICS transaction.

One of the following keywords can be provided to indicate what action IAM/PLEX will take when the owner of the record lock is other than a CICS transaction, such as a batch job or TSO user:

- **WAIT** specifies that the requestor of the lock will wait until the record lock is available when the owner is not a CICS transaction.
- **TIMEOUT** specifies that the requestor of the lock will wait for a short time as specified by the CICSTIMEOUT parameter (see below). If the lock is still not available, then the request will fail.
- **NOWAIT** specifies that the request for the record lock will be immediately failed when the owner is not a CICS transaction.

One of the following keywords can be provided to indicate whether IAM should fail the CICS transaction with an ABEND, or just fail the request following standard VSAM protocol with a logical error code:

- **ABEND** specifies that IAM will abend the CICS transaction. This requires the installation and activation of an IAM provided CICS Global User Exit, as described later. The transaction will abend with a code of AKC3.
- **NOABEND** specifies that IAM will fail the request with a logical error code.

The default values are: CICSLOCK=(CICSWAIT,TIMEOUT,NOABEND)

CICSTIMEOUT=

Specifies the maximum time, in seconds that a CICS transaction will wait for a record lock if CICSTIMEOUT or TIMEOUT has been specified for the CICSLOCK parameter. Default is 30 seconds, maximum is 9999.
**Parameter** | **Description**
---|---
DJB= | This parameter provides a way to control Dynamic Job Backout. The following values are accepted:

**YES** – Specifies that Dynamic Job Backout will be automatically invoked for any job steps that abend, and have recoverable IAM files accessed through IAM/PLEX. Changes made to recoverable IAM files for the abending job step will be removed from the affected files, back to either the most recent batch syncpoint taken by the job step, or if no syncpoints were taken, then to the beginning of the job step.

**NO** – Specifies that Dynamic Job Backout will not be automatically invoked, unless the failing job step has opened a recoverable IAM file through IAM/PLEX with the DJB=YES IAM override.

**DISABLED** – Dynamic Job Backout support is disabled, and will not be invoked even with a DJB=YES IAM override.

The default value is NO.

JRNBUFF= | Optional parameter that specifies the number of buffers that IAM/PLEX will use for journaling. Valid values are from 2 to 255. Low activity systems can use fewer buffers, while more active systems will benefit from additional buffers.

The default value is 32.
IAM/PLEX Parameters

**Parameter** | **Description**
--- | ---
**LOCK=** Specifies the action that IAM/PLEX will take when record lock contention is encountered for a requestor other than a CICS transaction, such as a batch job or TSO user. Three sub-parameters can be provided. The first instructs IAM/PLEX what action to take when the owner of the lock is a CICS transaction. The second instructs IAM/PLEX what action to take when the owner of the lock is other than a CICS transaction, such as a batch job or TSO user. The third tells IAM/PLEX that if it is going to fail the request due to lock contention is it to pass a logical error code back to the caller, or abend the job or TSO user.

One of the following keywords can be provided to indicate what action IAM/PLEX will take when the owner of the record lock is a CICS transaction:

- **CICSWAIT** specifies that the requestor of the lock will wait until the record lock is available when the owner is a CICS transaction.
- **CICSTIMEOUT** specifies that the requestor of the lock will wait for a short time as specified by the LOCKTIMEOUT parameter (see below), if the owner of the lock is a CICS transaction. If the lock is still not available, then the request will fail.
- **CICSNOWAIT** specifies that the request for the record lock will be failed if the owner is a CICS transaction.

One of the following keywords can be provided to indicate what action IAM/PLEX will take when the owner of the record lock is other than a CICS transaction, such as a batch job or TSO user:

- **WAIT** specifies that the requestor of the lock will wait until the record lock is available when the owner is not a CICS transaction.
- **TIMEOUT** specifies that the requestor of the lock will wait for a short time as specified by the LOCKTIMEOUT parameter (see below). If the lock is still not available, then the request will fail.
- **NOWAIT** specifies that the request for the record lock will be immediately failed when the owner is not a CICS transaction.

One of the following keywords can be provided to indicate whether IAM should fail the requestor with an ABEND, or just fail the request following standard VSAM protocol with a logical error code:

- **ABEND** specifies that IAM will abend the batch job or TSO user. The abend code will be a U0185.
- **NOABEND** specifies that IAM will fail the request with a logical error code.

The default values are: LOCK=(CICSWAIT,TIMEOUT,NOABEND)

**LOCKCHKPT=** Specifies the number of minutes between checkpoints on the lock recovery dataset. Valid values are 1 to 60. Refer to Section 25.06 for information on this parameter.

The default value is 5 minutes.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCKMEM=</td>
<td><strong>64BIT</strong> specifies that 64 bit storage will be used for the record lock table. If used a <strong>MEMLIMIT</strong> value may need to be specified on the EXEC statement in the IAM/PLEX proc at startup. If no 64-bit storage is available, then IAM/PLEX will default to using 31 bit storage for the record lock table. <strong>31BIT</strong> specifies that 31 bit storage will be used for the record lock table and is the default if no LOCKMEM parameter is provided or if no 64-bit addressable storage is available. Specifies the type of start up to be done for the Lock Recovery Process. Valid values are: <strong>COLD</strong> will result in no attempt to restore the record locks and force the reformatting of the lock recovery dataset. <strong>WARM</strong> will result in restoring the record lock environment for recoverable files to what it had been at the time when the IAM/PLEX region became unavailable. The default value is <strong>WARM</strong>.</td>
</tr>
<tr>
<td>LOCKTIMEOUT=</td>
<td>Specifies the time, in seconds, that a batch job or TSO user will wait for a record lock to become available if the CICSTIMEOUT and / or TIMEOUT value have been specified for the LOCK= parameter. The default value is 30 seconds, maximum is 9999.</td>
</tr>
<tr>
<td>LOGSTRM=</td>
<td>Specifies the name of the IAM log stream to be used for journaling, when the System Logger is going to be used. This value must match the log stream name defined by IXCMIAPU. This value is only used if the SYSLOGGER=YES parameter has been specified. In single system mode the default value, if not specified, is IAMLOGR.sysname, where sysname is the value provided at IPL time in the IEASYSxx member for SYSNAME. In SYSPLEX mode the default value is IAMLOGR.groupname, where groupname is the 8 character RLSGROUP name provided in the parmlib member when the IAM/PLEX address space starts. Using the System Logger is mandatory when using journaling with IAM/PLEX.</td>
</tr>
<tr>
<td>MAXBUFNO=</td>
<td>Specifies a value to be used as the default MAXBUFNO for IAM files opened under IAM/PLEX. If not specified, the default will be the normal IAM default from the IAM Global Options Table. Valid values are from 1 to 2048. This value may be overridden for specific files by an IAM ACCESS override cards that have been included with IAM/PLEX. If this value is changed when the parameters are reread by the CHANGEPARAM command, it will take effect for datasets opened subsequent to the change. Currently opened datasets will not have their MAXBUFNO value changed.</td>
</tr>
<tr>
<td>MAXIOTASK=</td>
<td>Specifies the maximum number of file I/O tasks to be permitted. This represents the maximum number of concurrent I/O requests that the IAM/PLEX address space will be able to be actively processing at any point in time. Additional I/O requests will be delayed pending availability of an I/O task to process it. Values from 8 to 255 can be specified. Default value is 64.</td>
</tr>
<tr>
<td>MAXJOBS=</td>
<td>Specifies the maximum number of jobs that can have IAM files opened under the IAM/PLEX address space. This is primarily used for setting up the record locking tables. Specifying too small of a value may cause increased overhead for the IAM lock manager if the actual number of jobs exceeds this limit. Values from 1 to 9999999 can be specified. Default value is 1024. Changing this value will only take effect when IAM/PLEX is restarted.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MAXLOCKS=</td>
<td>Specifies the approximate maximum number of record locks that are anticipated to be needed at any particular point in time. This value will be used to determine the size of the lock lookup hash table. Batch jobs will not be able to exceed holding this number of locks concurrently. Specifying too small of a value may cause increased overhead for the IAM lock manager. Values from 1 to 67108859 can be specified. When using LOCKMEM=31BIT the default value is 133103 and should not be made larger than 2129903. When LOCKMEM=64BIT the default is 2129903. Changing this value will only take effect when IAM/PLEX is restarted.</td>
</tr>
<tr>
<td>MAXTRANS=</td>
<td>Specifies the approximate maximum number of concurrently active transactions that a CICS system can have. Too small a value may cause increased overhead for the IAM lock manager. Values from 1 to 9999999 can be specified. Default value is 512. Changing this value through the CHANGEPARM command will take effect for CICS regions that are started subsequent to the change.</td>
</tr>
<tr>
<td>MESSAGES=</td>
<td>Specifies what messages should be printed to the IAM/PLEX RLSLOGDD file. Valid values are ALL, INFO, or ERROR. ALL is primarily intended for problem diagnosis, as various messages will be written indicating many frequent events. INFO will provide a detailed activity log, including when data sets are opened and closed. ERROR will limit the RLSLOGDD messages to only those issued in error situations. This value can be dynamically changed via operator command. Default value is INFO.</td>
</tr>
<tr>
<td>MINBUFNO=</td>
<td>Specifies a default minimum number of buffers per IAM file. Valid values are from 1 to 2048. The default value is 1, unless otherwise overridden by an IAMOVRID specification. If this value is changed when the parameters are reread by the CHANGEPARM command, it will take effect for datasets opened subsequent to the change. Currently opened datasets will not have their MINBUFNO value changed.</td>
</tr>
<tr>
<td>MINIOTASK=</td>
<td>Specifies the minimum, and also starting, number of tasks in the IAM/PLEX address space that will be used to process I/O requests. Values from 1 to 255 can be specified. Default value is 4, which is recommended with the changes that were made to the IAM/PLEX I/O task dispatching.</td>
</tr>
<tr>
<td>REMOTERLS</td>
<td>Specifies that this IAM/PLEX instance is a TARGET region only and cannot be connected to for routing open/close or I/O requests to other members of an RLSGROUP in the SYSPLEX. Note that if there is already an IAM/PLEX instance in the same RLSGROUP that is acting as the router, then the one being started will default to REMOTERLS.</td>
</tr>
<tr>
<td>RLSID=</td>
<td>Specifies the four character identifier for this IAM/PLEX instance. Each IAM/PLEX instance in a SYSPLEX will need to have its own unique identifier. If not specified, the value from the IAM Global Options Table will be used.</td>
</tr>
<tr>
<td>RLSGROUP=</td>
<td>Specifies the 8 character identifier for the RLSGROUP the IAM/PLEX instance is to connect to. If not specified then IAM/PLEX will run in a single system or non SYSPLEX mode, equivalent to an IAM/RLS instance.</td>
</tr>
<tr>
<td>STATS=</td>
<td>When specified as YES, the RLSINFO reports will be generated automatically every 30 minutes if any stats have changed during that time period. The default is NO. The report can always be generated manually via operator command. The RLSINFO DD statement must also be present in the IAMRLS PROC used to start the IAM/PLEX address space.</td>
</tr>
</tbody>
</table>
IAM/PLEX Parameters

Examples of IAM PARMLIB Parameters

Example of parmlib input for IAM/PLEX address space are shown below:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSLOGGER=</td>
<td>Specifies either YES or NO, to indicate whether the z/OS System Logger is to be used instead of the sequential data sets. This option is required when the RLSGROUP parameter is provided and journaling is desired in the SYSPLEX environment. Refer to the Section 25.30 on journaling for more information on using this capability. If YES is specified, you can also specify the LOGSTRM= parameter, indicating the name of the log stream. If no LOGSTRM is specified the log stream name will default to IAMLOGR.systemname in single system mode or to IAMLOGR.rlsgroupone name in SYSPLEX mode. When specified as YES, all of the LOGDSNx parameters are ignored. Default value is NO.</td>
</tr>
<tr>
<td>SYSOUT=</td>
<td>Optional parameter specifying the SYSOUT class to be used for printed output directed to the RLSLOGDD. This parameter is ignored if an RLSLOGDD DD card is in the IAM/PLEX started proc. If an RLSLOGDD DD card is not included in the IAM/PLEX proc, then the SYSOUT class specified by this parameter will be used. If no value is specified, then the default SYSOUT class (equivalent to SYSOUT=*) for the started task will be used. Users can spin off the RLSLOGDD SYSOUT with the IAM/PLEX LOGSWITCH command.</td>
</tr>
</tbody>
</table>

MINBUFNO=32, MAXBUFNO=300, MAXIOTASK=64, MINIOTASK=4
MESSAGES=INFO
CICSTIMEOUT=(CICSTIMEOUT,TIMEOUT,NOABEND)
LOCK=(CICSWAIT,TIMEOUT,ABEND)
CICSTIMEOUT=10
LOCKTIMEOUT=90
SYSLOGGER=NO
SYSOUT=X
JRNBUFF=128
RLSID=RLS1 Set RLSID of RLS1 for this IAM/PLEX
RLSGROUP=RLSGRP01 Connect to RLSGROUP: RLSGRP01
LOCKCHKPT=5 Record lock recovery checkpoint 5 minutes
LOCKSTART=WARM Perform lock recovery on startup
LOCKMEM=64BIT Use 64-bit storage for record locks
STATS=ON

Figure 25-5: Example of IAM/PLEX Parmlib Parameters. Member PLXPRM1 in ICL
Abending CICS Transactions
If ABEND was specified for the CICSLOCK parameter, then to cause the abends to occur the IAM provided CICS Global User Exit (GLUE) must be activated. This IAM provided Global User Exit (GLUE) will purge transactions that receive an indication from IAM/PLEX to do so. This indication will only be presented when the action specified for the IAM/PLEX CICSLOCK parameter is set to ABEND. The transaction will abend with a code of AKC3. For information on setting this up, refer to Section 25.50, CICS Considerations.

IAM Overrides
IAM ACCESS Overrides can be specified for the IAM/PLEX instance. The overrides can be a member of a card image PDS, such as the IAM ICL. Some of the IAM ACCESS overrides that may be useful to IAM/PLEX are BUFSPACE, CONSISTENT, JRNAD, MINBUFNO, MAXBUFNO and REREAD. The IAM CREATE overrides are not relevant to IAM/PLEX because no file creations will take place under the IAM/PLEX instance.

The BUFSPACE, MINBUFNO or MAXBUFNO can be used on a dataset by dataset basis, to either use more or less buffering than what is being defaulted to under IAM/PLEX. The need to alter the buffering values should be minimal. This is because IAM/PLEX offers a MAXBUFNO value to be specified as a default for all files opened under IAM/PLEX as a start up parameter that was previously described.

The JRNAD override can be used to enable journaling for all IAM files accessed under IAM/PLEX, by providing an ACCESS DD=&ALLDD,JRNAD=BOTH. Be sure to specify JRNAD on any additional ACCESS override cards provided for specific data sets, as they will not pick up the value from the &ALLDD override. Using the override may be the preferred way to control journaling for IAM/PLEX, rather than on an individual file basis. The journaling can be turned off for specific datasets by an additional ACCESS override indicating the dataset name.

If you are using the IAM overrides with IAM/PLEX, then you will most likely want to specify the REREAD keyword on each of the override cards. Specification of REREAD will cause IAM to read the IAM overrides every time an IAM file is opened, rather than only for the first dataset opened, and saving the results in storage. While this is additional overhead, it will allow you to change the overrides without having to stop and restart IAM/PLEX for the new overrides to take effect.

Shown below is an example of overrides for an IAM/PLEX instance:

```
ACCESS DD=&ALLDD,INDEXSPACE=64BIT,REREAD
ACCESS DD=RLOKJRNL,INDEXSPACE=64BIT,NORLS,JRNAD=NONE,REREAD
```

Figure 25-6: Example of IAM/PLEX Override Parameters. Member PLXOVRID in ICL
IAM/PLEX Automatic Dataset Eligibility

IAM/PLEX Data Set Selection

IAM can automatically decide during OPEN processing which IAM data sets are to be handled by IAM/PLEX. Automatic selection is based on the share options that a data set was defined with, if an RLSID was specified when a dataset is defined or loaded, and/or by the data set name table. An installation specifies the criteria IAM is to use through the IAM Global Options Table, via the RLS option. Using the RLS option, users can specify the minimum share option required for eligibility for IAM/PLEX processing, and whether or not a data set name table has been provided to IAM/PLEX. If a combination of share option and data set name eligibility is requested, the user can specify via the RLS option whether a data set must meet both the share option criteria and be in the data set name table, or if a match on either one sets the data set as eligible for IAM/PLEX. The values that can be specified for the RLS Global Option are shown below:

To minimize OPEN overhead in determining a dataset’s eligibility for IAM/PLEX processing consider implementing a process that specifies the RLSID for datasets as they are defined. The RLSID can be modified or removed through a DEFINE RECATALOG process when necessary.

RLS Global Option

Below is the description of the RLS Global Option. It continues on the next page.

<table>
<thead>
<tr>
<th>RLS Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>Indicate that to be automatically eligible for IAM/PLEX, a data set must have the specified share option AND must be selected in the dataset name table.</td>
</tr>
<tr>
<td>NONE</td>
<td>Indicates that no data sets are to be considered for automatic eligibility for IAM/PLEX. Set this option if you are not going to be activating the IAM/PLEX address space, or if you want to manually direct activity to IAM/PLEX through the IAM Overrides. This is the default option as shipped.</td>
</tr>
<tr>
<td>OPTIONAL</td>
<td>Allows data sets to be opened without IAM/PLEX when the IAM/PLEX address space is not active. This is useful for when the TABLE option has been specified and IAM/PLEX may not be active on all LPAR’s because IAM/PLEX has to at least have been active at some point in time for the dataset name table to be loaded. If specified, this value will also be applied to datasets defined with an RLSID, in that if IAM/PLEX is not active, the dataset will be opened locally.</td>
</tr>
<tr>
<td>OR</td>
<td>Indicates that to be automatically eligible for IAM/PLEX, a data set must have either the specified share option or be selected in the dataset name table. <strong>If a data set meets the share option eligibility, then the data set name table will not be examined.</strong> If a data set does not meet the share option eligibility criteria, then it must be selected in the dataset name table to be eligible for IAM/PLEX processing. OR is the default if both TABLE and a SHAREx value are specified.</td>
</tr>
<tr>
<td>REQUIRED</td>
<td>Indicates that IAM/PLEX MUST BE ACTIVE if a data set is eligible for IAM/PLEX processing, or when the TABLE Global Option is specified. If IAM/PLEX is not active, then any attempt to open a potentially IAM/PLEX eligible data set will be failed. REQUIRED is the default. For datasets defined with an RLSID, IAM/PLEX must be active even if RLS=NONE is set, unless RLS=(NONE,OPTIONAL) is specified.</td>
</tr>
<tr>
<td>SHARE1</td>
<td>Indicates that data sets with any cross-region share option value (1, 2, 3, or 4) will be eligible for IAM/PLEX processing. Use of this value is not recommended.</td>
</tr>
<tr>
<td>SHARE2</td>
<td>Indicates that data sets with cross-region share options of 2, 3, or 4 will be eligible for IAM/PLEX processing. Use of this value is not recommended.</td>
</tr>
</tbody>
</table>
### Using the RLS Global Option

The following discussion applies to the use of the RLS Global Option. Full use of the IAM Global Option Change Facility is described in Section 91 of the IAM Users Manual. Multiple values can be specified for the RLS option, by enclosing them within parenthesis. When the RLS option is specified on the ZAP control statement, the values specified will act as a complete replacement for the existing RLS values they do not act in combination with the prior values. As shipped, the default value is `RLS=NONE`, meaning that no IAM files are automatically eligible for IAM/PLEX processing.

You might desire some combination of selection criteria; for example, you want all share option 3 or 4 files to be eligible, and some selected share option 2 files. For this, you would specify `RLS=(SHARE3,OR,TABLE)`. You would then specify the name(s) of the other files to be included or excluded in the data set name tables. The reason for both an include and an exclude list is so that you could select all data sets beginning with `PROD.SHARE` for IAM/PLEX processing. You would specify in the data set name table `SELECT DSN=PROD.SHARE*`. However, perhaps there is a file with that name prefix that you want excluded from IAM/PLEX, so you could specify `EXCLUDE DSN=PROD.SHARE.FILE1`.

Another possibility is that you might want only share option 3 or 4 files to be eligible for IAM/PLEX, however you might want to exclude some selected files. In this situation, you would specify `RLS=(SHARE3,AND,TABLE)`. Then in the data set name table, you would specify `SELECT DSN=*` which will include all data sets, and then exclude specific data sets, such as `EXCLUDE DSN=PROD.DONOT.SHARE.FILE1`.

### RLS DSN Table

The data set name table is provided to the IAM/PLEX address space through the `IAMDSNTB` DD statement. This DD must specify a card image data set, either sequential or a PDS member. The table is read in during initialization of the first IAM/RLS or IAM/PLEX address space started on an LPAR, and stored in ECSA. The table is kept in ECSA storage to minimize the overhead of searching the table during IAM OPEN processing. Customers can update the data set name tables within the IAMDSNTB data set, and activate the updated tables by using the `CHANGEDSNT` command of the IAM/PLEX address space. If you have IAM/PLEX running on multiple LPARs with the same RLSGROUP then the command is automatically sent to those address spaces.

<table>
<thead>
<tr>
<th>RLS Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHARE3</td>
<td>Indicates that data sets with cross-region share options of 3 or 4 will be eligible for IAM/PLEX processing.</td>
</tr>
<tr>
<td>SHARE4</td>
<td>Indicates that only data sets with a cross-region share option of 4 will be eligible for IAM/PLEX processing.</td>
</tr>
<tr>
<td>TABLE</td>
<td>Indicates that IAM is to search the dataset name table to determine eligibility for IAM/PLEX processing, subject to other criteria. Be careful of using the options TABLE with REQUIRED. With both of those options, no IAM data sets can be opened until an table has been loaded into virtual storage. If IAMRLS is not started up immediately during the IPL process, consider starting the IAMRLS with PARM=DSNTLOAD which will start VIF and load the DSN TABLE then exit. The default is that the data set name tables will not take part in eligibility selection for IAM/PLEX.</td>
</tr>
</tbody>
</table>
Within the IAMDSNTB data set, two lists of data set names can be provided. Each data set name entry is considered a full data set name unless it contains 1 or more of the special masking characters. The masking characters allow a single entry in the table to represent multiple dataset names, resulting in a smaller table. Refer to the topic below on dataset name masking for complete information. Enhanced dataset name masking is supported in IAM V9.2, however the old use of a trailing single asterisk (*) will still work the same as it did previously.

The first list is the data set name selection list. It is specified by "SELECT RLSID=rlsx,DSN=" followed by a list of data set names, separated by commas. All of the datasets included on that select statement are assigned to the specified RLSID. Multiple cards can be used, however each data set name or mask must be fully contained within a single card image. A closing parenthesis is used at the end of the last data set name. If there is just one data set name, or only a * (single asterisk) then no parenthesis are needed.

The exclude list is specified by the “EXCLUDE DSN=” followed by a list of data set names, separated by commas. Multiple cards can be used however each data set name or mask must be fully contained within a single card image. A closing parenthesis is used at the end of the last data set name. If there is just one data set name, or only an *, then no parenthesis are needed. The EXCLUDE statement must start on a new card.

The EXCLUDE list is always searched first, so if the dataset name matches any mask in the exclude list it is automatically not eligible for IAM/PLEX processing even if it is in the INCLUDE list. As much as possible, it is recommended that the EXCLUDE list be kept short but very broad so that as many datasets as possible will be recognized as not requiring IAM/RLS or IAM/PLEX processing.

You can group dataset names by RLSID. If no RLSID is specified on the SELECT statement then it will be blank in the table and during open processing the RLSID provided by the CREATE or ACCESS overrides or the default in the Global Options Table will be used. There can be multiple SELECT DSN= and EXCLUDE DSN= statements in the IAMDSNTB DD input file. Each statement will be processed with the dataset names added to the corresponding Select or Exclude table in storage. If an error is encountered on the first SELECT or EXCLUDE statement, and there was already a Select or Exclude table in storage, then the old table will still be in effect. If an error is encountered on a subsequent SELECT or EXCLUDE statement in the processing of the IAMDSNTB DD input file then the in storage Select and Exclude table will contain only those dataset names that were successfully processed on the previous input cards before the error was encountered. A maximum of 399 input cards can be specified for each SELECT or EXCLUDE statement with multiple dataset names on each card. A total maximum of 32,767 names can be specified for each list. However, it is highly recommended that you keep the lists as short as possible, to keep the ECSA storage requirement low, and to minimize the searching CPU time. For additional information on acceptable formats, refer to Section 01.04 of the IAM Users Manual for general information on control statement format.
To provide more flexibility and help reduce the size of the table, IAM provides a masking facility that can be used when specifying dataset names on the SELECT and EXCLUDE statements. The mask may contain:

- Any valid alphanumeric or national character representing itself
- / (slash) symbol represents a single valid character
- | (vertical bar) represents a single alphabetic character
- + (plus) represents a single valid numeric character
- ? (question mark) represents a single valid national character i.e. (#, $, or @ in US, or € in Europe)
- * (single asterisk) represents zero or more valid characters within one index level except when it is the last character in the mask in which case it is treated as meaning any subsequent characters.
- ** (double asterisk) represents zero or more valid characters contained in one or more index levels (including their periods)
- . (period) represents a period (index level) in the dataset name except as noted in special cases below. A period at the beginning of the mask, or immediately following another period is interpreted as an *. (single asterisk, period).
- **. (double asterisk, period) at the beginning of a string represent one or more index levels at the beginning of the dataset name
- .* (period, asterisk) at the end of the string represents one or more index levels at the end of the dataset name
- .**. (period, double asterisk, period) in the middle of the string represents either a SINGlE period, or one or more index levels at that point in the dataset name
- The last character specified cannot be a period.
- *** (triple asterisks) are invalid.
- An **.** is invalid, If you want to represent at least a specific number of index levels use *.**
Below is a sampling of some of the dataset name variations that can be built with the masking capability.

- **ABC.GLEDGER.MASTER**  This example is for a complete dataset name.
- **XYZ.PROD.GROUP*.MSTR.MUPD**  This example allows for multiple datasets with additional characters as part of the GROUP index level. For example this would pick up GROUP0, GROUP1A, GROUP001, GROUPABC, etc.
- **XYZ.PROD.GROUP++.MSTR.MUPD**  In this variation the GROUP must be followed by 2 numeric digits, such as GROUP25, or GROUP99, etc...
- **PROD.IAM.PAYROLL.***  All datasets that begin with PROD.IAM.PAYROLL.
- **XYZ.*.SHARED**  Allows datasets that start with XYZ, have any second level index, and ending with.SHARED, for example: XYZ.F1.SHARED or XYZ.ABC12345.SHARED
- **PROD//.ARMMASTER.***  Includes datasets that begin with PROD followed by any two valid characters such as PRODNY.ARMMASTER.abc which are followed by any characters at the end. Other possible matches include: PRODNJ.ARMMASTER.d060510.master, or PROD99.ARMMASTER.xyz
- **TEST***  Any datasets that begin with the characters TEST.

Shown below is an example of an RLS Data Set Name table:

```
SELECT RLSID=RLS1,DSN=(PROD.PAYROLL.IAM.*,
                       PROD.AR///.IAM.*, 
                       IAMV.**.RLS, 
                       PROD.APAY.*.SHARED)

SELECT RLSID=RLS2,DSN=(PROD2.PAYROLL.IAM.*,
                       PROD2.AR*.IAM.*, 
                       IAMV2.*.RLS)

EXCLUDE DSN=(PROD.PAYROLL.IAM.TESTFILE,
             PROD.AR.IAM.HISTORY,
             TEST.*, 
             IAMV.RLS999.DONOT.SHARE.*)
```

*Figure 25-7:  Example of IAM/PLEX Data Set Name Table Member PLXDSNTB in ICL*
The implementation method chosen by many users is the dataset name table. This method offers a single point of control over which datasets are going to be utilizing IAM/PLEX and eliminates the need for any JCL changes for jobs utilizing IAM/PLEX. The requirements to utilize this method are:

1. Building a card image dataset containing the dataset SELECT list and if necessary a dataset EXCLUDE list along with the desired RLSID to handle each SELECT group.
2. Specification of the table via a DD statement in each of the IAM/PLEX and IAM/RLS JCL procedures. Each IAM/PLEX and IAM/RLS procedure MUST specify the same table.
3. Revising the IAM Global Options to RLS=(TABLE,OPTIONAL) or RLS=(TABLE,REQUIRED).
4. Starting an IAM/PLEX (or IAM/RLS) address space on each LPAR that is using IAM files during IPL, or before any IAM files are processed. If RLS=(TABLE,REQUIRED) is used, then no IAM files will be able to be opened until an IAM/PLEX or IAM/RLS address space has initialized a table in virtual storage.

You specify the RLSID on the SELECT statement like this:

- SELECT RLSID=RLS1,DSN=(IAM.DATASET.ONE, IAM.DATASET.TWO,IAM.APPL1*)
- SELECT RLSID=RLS2,DSN=(IAM.DATASET.THREE,IAM.DATASET.FOUR)

If the RLS Global Option is set to use the dataset name table, then the first 2 datasets and any that match the beginning of the file name IAM.APPL1 will automatically be opened under RLS1, and the second 2 datasets will be opened under RLS2.

A second recommended method of implementing the use of multiple IAM/PLEX address spaces is to have all files that are going to be using IAM/PLEX services to be defined with the IAM CREATE Override of RLSID=nnnn indicating the IAM/PLEX address space is to process it. The primary advantage of this method of implementation is that it eliminates the overhead during open processing of every IAM file for scanning the table to determine eligibility for IAM/PLEX and eliminates the need for the ECSA storage used to hold the table. There also is no need to change the JCL of any of the job(s) using this dataset, other than the job that defines or reorganizes it. The specification of the CREATE Override of RLSID will indicate that this dataset is requesting / requiring IAM/PLEX services from the specified IAM/PLEX address space. The RLSID can be revised or eliminated at any time through the use of the DEFINE RECATALOG process.
Alternative Implementations

An alternative implementation method is to use the IAM ACCESS Overrides. Using this method, the jobs and CICS regions that have the datasets requiring other than the default IAM/PLEX address space will need to have their JCL changed to include an IAM ACCESS Override, for example:

```
//IAMOVRID DD *
 ACCESS DD=&ALLDD,RLSID=xxxx
 ACCESS DSN=IAM.DATASET.SHARE1,RLS,RLSID=xxxx
 ACCESS DSN=IAM.DATASET.SHARE2,RLS,RLSID=yyyy
/*
```

*Figure 25-8: Example of using ACCESS overrides to direct datasets to IAM/PLEX*

The following notes explain the above example:

- The first override with DD=&ALLDD will in effect set the default RLSID for any IAM datasets that require IAM/PLEX usage that do not have explicit overrides.
- The second override requests IAM/PLEX processing for dataset.SHARE1. Because this is an explicit override, none of the information from the DD=&ALLDD applies, so the RLSID needs to specified here even though it is the same as the one on the DD=&ALLDD card.
- The third override card specifies IAM/PLEX processing for file SHARE2, but it is directed to a different IAM/PLEX address space, which must be a member of the same RLSGROUP as the other IAM/PLEX address space.

If there are overrides for additional specific datasets, then they will have to include the RLSID=xxxx override as well.
IAM/PLEX provides an optional facility to journal before and/or after images of file updates to IAM files being processed by IAM/PLEX. These journals can be used to aid in data recovery due to either hardware or software failures. For example, if a media failure occurs and an IAM data set has to be restored, the IAM/PLEX journals can be used to reapply the updates that had been made to the file since it was backed up to bring the file status up to the point of failure. If before images are captured then IAM can backout updates performed by failing job steps. The IAM journaling is not a replacement for CICS journaling and CICS dynamic transaction backout, but rather an additional facility that provides for increased recovery capabilities. IAM/PLEX as a default, will not journal the before images for updates being done by CICS transactions, because CICS will handle the transaction backouts. However, this can be changed if an installation needs to have those records included in the IAM journals.

IAM provides programs to perform recoveries, IAMJREST and IAMBREST. IAMJREST runs as a batch job that can perform forward or backout recoveries. IAMBREST provides for dynamic job step backout of updates performed by a batch job step that abends. IAM also provides a program to perform utility functions on the IAM/PLEX logstreams, called IAMJUTIL. IAMJREST and IAMJUTIL can also be used to process journals created by IAM outside of IAM/PLEX control, as described in Section 10.88 IAM Journal and Recovery. Customers can provide their own utility program to process the data contained in the journals, and to even perform recoveries if so desired. For information on the format of the data contained in the IAM/PLEX logstream records, contact Innovation.

When implementing IAM/PLEX one must use the z/OS system logger so that all IAM/PLEX members of an RLSGROUP from different LPARS are logging to the same journal datasets. The type of journaling and the logstream files are specified by the IAM/PLEX startup parameters SYSLOGGER, and LOGSTRM, which are described in Section 25.10.

Customers select the IAM data sets that they want to have IAM journal, by one of the following methods:

- Specification of the IAM ACCESS Override JRNAD on the IAM/PLEX startup procedure, for example: ACCESS DD=&ALLDD,JRNAD=BOTH.
- Specification of the IAM CREATE Override JRNAD when defining the IAM data set(s) that require journaling.
- Specification of the LOG parameter on the IDCAMS DEFINE for DFSMS managed IAM data sets.
IAM/PLEX Journaling

System Logger

IAM/PLEX must use the z/OS system logger for journaling file updates to allow for back out and forward recovery of IAM files. The z/OS system logger must be activated and configured. For more information on the z/OS system logger see the following publications:

z/OS MVS Setting Up A Sysplex

z/OS MVS Initialization and Tuning Reference

Perform the following tasks in order to set up the IAM System Logger environment for IAM/PLEX. Your z/OS systems programmer must do some of these tasks.

1. Make sure you have authorization to the z/OS system logger address space and that the z/OS system logger (IXGLOGR) is running before you define and use IAM log streams. The section “Define Authorization to System Logger Address Space” in the z/OS MVS Setting Up a Sysplex manual provides more information about this.

2. You must also have authorization to the z/OS IXCMIAPU utility. This utility is used to define, update, and delete entries in the LOGR couple data set. IXCMIAPU is documented in the z/OS MVS Setting Up a Sysplex manual in appendix B.

3. Use the IXCL1DSU utility to define and format the LOGR couple data set. Your installation may have already done this if they are using the z/OS system logger for other products. This utility is documented in the z/OS MVS Setting Up a Sysplex manual.

4. Define the Structure in the CFRM policy for the SYSPLEX. This is the structure in the coupling facility that the SYSTEM Logger will initially write the data in before it is moved to logger datasets. This is also done using the IXCMIAPU utility.

5. Define the Structure in the LOGR policy using IXCMIAPU utility.

6. Define the IAM log streams with the DASDONLY(NO) and the STRUCTNAME( ) parameters in the LOGR couple data set using the z/OS utility IXCMIAPU. See the section "Add Information about Log Streams and Coupling Facility Structures to the LOGR Policy" in the z/OS MVS Setting Up a Sysplex manual. The following JCL can be used to set the values required for the IAM log stream using the IXCMIAPU utility.

7. Define the Logger address space (started task) as a TRUSTED facility to avoid RACF problems with the log datasets and structures.
The following statements need to be added to the CFRM policy.

```
/* */
/* IAMSTR01= IAMLOGR STRUCTURE */
/* */
STRUCTURE NAME(IAMSTR01)
  SIZE(60000)
  INITSIZE(60000)
  PREFLIST(PARTCF)
  ALLOWAUTOALT(NO)
```

*Figure 25-9: Example of IAMLOGR Structure, Member IAMSTR01 in ICL*

The following example JCL shows how to build the IAMLOGR structure.

```
//*
//* JCL TO DEFINE THE IAMLOGR STRUCTURE IN THE LOGR POLICY
//*
//*
//STEP1 EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
DATA TYPE(LOGR) REPORT=YES)
  DEFINE STRUCTURE NAME(IAMSTR01)
    LOGSNUM(10)
    AVGBUFSIZE(4096)
    MAXBUFSIZE(65532)
/*
```

*Figure 25-10: JCL to Define IAMLOGR Structure: Member IAMLGSTR in ICL*
Define an IAM Log Stream

The following JCL example shows how to create the actual log stream for an IAM/PLEX group.

```java
//STEP1     EXEC  PGM=IXCMIAPU
//SYSPRINT  DD    SYSOUT=*  
//SYSIN     DD    *  
    DATA TYPE(LOGR) REPORT(YES)
    DEFINE LOGSTREAM NAME(IAMLOGR.RLSGRP01)
       STRUCTNAME(IAMSTR01)
       DESCRIPTION(IAM)
       DASDONLY(NO)
       STG_STORCLAS(TSTDATA)
       LS_STORCLAS(TSTDATA)
       LS_DATACLAS(RRS1)
       STG_SIZE(48000)
       LS_SIZE(48000)
       HLQ(SYSPXCF)
       LOWOFFLOAD(00)
       HIGHOFFLOAD(50)
       AUTODELETE(YES)
       DIAG(YES)
       RETPD(10)
```

*Figure 25-11: Example of JCL to Define Log Stream: Member IAMLGDEF in ICL*

The LOWOFFLOAD(00) and the HIGHOFFLOAD(50) should not be changed. If you set the HIGHOFFLOAD too high the System Logger could reject logging requests while it is offloading data resulting in a disabling of the journaling facility. The RETPD(10) is the number of days you want the data kept before it can be deleted from the logstream and its offload datasets. The IAMJUTIL utility can also be used to copy data from the logstreams to other datasets for archiving purposes.

Handling Errors

When there is an error, the IAM/PLEX JRNAD exit will end up waiting on the buffer queue until buffers are emptied and the request on top of the queue gets posted when the error is resolved. At this time the only error automatically handled is for reason code of 00000860 indicating that an offload is needed. All other tasks trying to journal are stacked up behind the one on the buffer queue waiting for the journal buffer lock.

What can be done if there is a catastrophic error with the logstream, or coupling facility is the following:

1. Disconnect from the current logstream: F iamplex1,LOGGER,DISCON
2. Define a new logstream
3. Change the IAM/PLEX parameter library LOGSTREAM value.
4. Issue the IAM/PLEX CHANGEPARM command: F iamplex1,CHANGEPARM
5. Reconnect to the new logstream: F iamplex1,LOGGER,RECON
6. At that point the waiting request should be posted. If not, then issue the IAM/PLEX LOGGER POST command: F iamplex1,LOGGER,POST

- 25-37 -
**25.40 IAM/PLEX Operator Commands**

Certain aspects of the IAM/PLEX address space can be controlled by an operator using the 
**z/OS modify (F) command. Some of the functions include shutting down the IAM/PLEX address space, causing the IAM/PLEX address space to schedule a dump to a SYS1.DUMPxx data set, activating and stopping the IAM/PLEX trace facility, and applying PTF's to an active IAM/PLEX address space. The general format of the command is as follows:**

```
F iamplex address space name,command
```

For example, to dump an IAM/PLEX address space started with the name of IAMPLEX, enter the following:

```
F IAMPLEX,DUMP
```

The following IAM/PLEX commands can be used. Please note that some of the commands will prompt you to enter additional information via a WTO R. Also note that where a recognized abbreviation exists for a command or for operand it is shown by the underlined portion of the full value.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLY,dsn</td>
<td>This command is used to initiate application of Innovation supplied PTF's to the modules loaded in the IAM/PLEX address space storage. This facility enables customers to apply most maintenance without stopping and restarting IAM/PLEX processing. For dsn, specify the name of a sequential or partitioned dataset containing the fix to be applied. (For PDS's, include the member name in parenthesis.) For example:</td>
</tr>
<tr>
<td></td>
<td>F iamplex1,APPLY,IAMSYS.RLS.ICL(P910001)</td>
</tr>
<tr>
<td>CHANGEDSNT</td>
<td>This command will update the list of dataset names to be included and/or excluded from IAM/PLEX processing from the IAMDSNTB DD statement. This provides customers with a mechanism for picking up an updated list without having to stop and restart IAM/PLEX. The command will be transmitted to all of the members connected to the IAM/PLEX group.</td>
</tr>
<tr>
<td>CHAN GEPARM</td>
<td>This command will update the current IAM/PLEX parameters from the dataset and member specified on the IPARMLIB DD statement. While most parameters can be dynamically changed, some parameters will only effectively be changed by a restart of IAM/PLEX. For example, because MAXLOCKS is used to set an internal table size established at startup, changing that value will not have any effect until IAM/PLEX is restarted.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CLOSEFILE</td>
<td>This command will close the specified data set. It is intended to be used only in emergency circumstances to recover from a problem. Datasets are normally closed by the applications using IAM/PLEX. The file should first be closed by the application or CICS region before attempting to issue this command. Additional operands, separated by commas, are:</td>
</tr>
<tr>
<td></td>
<td><strong>DSN</strong>= Specifies the data set name.</td>
</tr>
<tr>
<td></td>
<td>One of the following is also required:</td>
</tr>
<tr>
<td></td>
<td><strong>JOBNAME</strong>= Specifies the jobname for which the dataset is to be closed.</td>
</tr>
<tr>
<td></td>
<td><strong>JOBID</strong>= Specifies the JES JOBID that identifies the job for which the data set is to be closed.</td>
</tr>
<tr>
<td></td>
<td><strong>ALL</strong> Specifies that the file is to be closed for every job that has it opened under IAM/PLEX.</td>
</tr>
<tr>
<td></td>
<td>For Example: F iampelx1,CL,DSN=my.open.ds,J=myjob</td>
</tr>
<tr>
<td></td>
<td>F iampelx1,CL,DSN=my.open.ds,ALL</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>This command, along with associated sub-parameters, will display requested information on the output file RLSLOGDD. The RLSLOGDD is used as a default because the output from the DISPLAY could be quite substantial. By adding the keyword ',CON' to the end of the DISPLAY request, the information will be displayed on the operator's console. One of the following parameters must be provided to indicate what information is to be displayed.</td>
</tr>
<tr>
<td></td>
<td><strong>CONTENTION</strong> – Requests IAM/PLEX to display any record locks held that have contention (other requestors waiting for the lock).</td>
</tr>
<tr>
<td></td>
<td><strong>EXCLUDETB</strong> – Indicates that the data set name exclusion table is to be displayed.</td>
</tr>
<tr>
<td></td>
<td><strong>GROUP</strong> – Displays all members connected to the RLSGROUP to which this IAM/PLEX is a member. The status and CPUID where each member is running is displayed.</td>
</tr>
<tr>
<td></td>
<td><strong>RETAINEDLOCKS[,DSN]</strong> – Indicates that information on the jobs, and optionally which data sets used by those jobs, have had locks retained due to an abend will be displayed.</td>
</tr>
<tr>
<td></td>
<td><strong>SELECTTB</strong> – Indicates that the data set name selection table is to be displayed.</td>
</tr>
</tbody>
</table>

For example: F iampelx1,DISPLAY,RETAINEDLOCKS  
F iampelx1,DISPLAY,CONT  
F iampelx1,DISPLAY,SELECTTB
IAM/PLEX Operator Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| DJBTRACE=   | This command is used to activate or deactivate the Dynamic Job Backout tracing. The Dynamic Job Backout trace is useful for determining exactly what file activity is being performed by Dynamic Job Backout. This can be used to verify proper operation, or for problem diagnosis relating to the recovery processing. When the trace is activated, by specifying DJBTRACE=YES, the next time a backout is attempted, the backout will dynamically allocate the RLSDJBDD file to a SYSOUT data type of dataset. When deactivated, by DJBTRACE=NO, upon completion of a backout, the output for the trace will be closed and de-allocated. Valid values are:  
  YES – Activate IAM/PLEX Dynamic Job Backout tracing.  
  NO – Deactivate IAM/PLEX Dynamic Job Backout tracing. |
| DUMP        | This command is used to request that the IAM/PLEX address space take a dump to a SYS1.DUMPxx data set. The dump will automatically include the data space containing the IAM Index Space. When a dump has been requested, this is the preferred method of obtaining the dump rather than the z/OS DUMP command, as this will automatically include the necessary storage areas for problem diagnosis. |
| JSWITCH     | This command causes logging to switch from the currently active log data set to the next log data set. In SYSPLEX mode the JSWITCH command can be used to start offload processing of the SYSTEM LOGGER journal data to an offload dataset. |
| LOGSWITCH   | This command causes IAM/PLEX to close and de-allocate the currently active file being used for output directed for the IAM/PLEX log (RLSLOGDD). Another SYSOUT dataset will be dynamically allocated using the SYSOUT class specified in the IAM/PLEX startup parameters. If an actual RLSLOGDD DD card is in the IAM/PLEX startup proc, then this command is ignored. |
| LOGGER      | These are some commands to help deal with any problems with the System Logger. What can be done here is that if there is a catastrophic error with the logstream, or coupling facility, they can disconnect from the current logstream, define a new logstream, change the RLS parmlib, run the changeparms command, then reconnect to the new logstream. At that point BJRNL should get posted by the SRB to recover, but just in case they can issue the POST command to get things started. There are 4 different parameters:  
  DISCON Used to disconnect from the current System Logger logstream  
  RECON Used to reconnect to the System Logger logstream  
  OFFLOAD Used to start an offload for the current logstream  
  POST Post the journal subtask to start writing log records again  
  For Example: F iamplex1,OFFLOAD |
| MESSAGES=   | Specifies what messages should be printed to the IAM/PLEX RLSLOGDD file. Valid values are ALL, INFO, or ERR. ALL is primarily intended for problem diagnosis, as various messages will be written indicating many frequent events. INFO will provide a fairly detailed activity log, including when data sets are opened and closed. ERR will limit the RLSLOGDD messages to only those issued in error situations. |
QUIESCE [,FORCE]

This command requests that the IAM/PLEX address space shut down. If FORCE is not specified, then when all of the currently open IAM files are closed, the IAM/PLEX address space will terminate. Any new open requests will be rejected.

If the FORCE keyword is specified, then all of the files will be immediately closed, and the address space will terminate normally. All subsequent requests by the jobs that were processing data sets through the IAM/PLEX address space will be rejected.

RELEASELOCKS

RELLOCKS

Specifies that IAM/PLEX is to release locks for the specified job or CICS transaction. Either retained or held locks may be released. Additional parameters may also be required to further identify the specific job step if there are multiple matches on the job name. The information necessary to provide the details required by this command can be obtained from the DISPLAY,RETAINEDLOCKS command. The following subparameters can be specified:

JOBNAME= specifies the 1 to 8 character job name.

JOBID= Specifies the 8 character JES jobid, for example: JOB12345

ASID= Specifies the hexadecimal digits of the address space ID number.

STEPNAME= Specifies the 1 to 8 character step name, and is optional.

TRNAME= Specifies the 1 to 4 character CICS transaction name.

TRID= Specifies the CICS transaction number, in 1 to 5 decimal digits.

For Example: F iamplex1,REL,J=myjob,ID=JOBnnnnn,S=step1

RESTART

This command will reverse a previously issued QUIESCE command that was issued without the FORCE keyword. Normal processing will be resumed by the IAM/PLEX address space, and new open requests will be honored. This will only function if the IAM/PLEX address space has not already terminated.

RESTORE,dsn

This command can be used to reverse the application of an Innovation supplied PTF from the modules loaded within the IAM address space. This can be used when a PTF has been applied using the APPLY command, and is either causing additional problems, or if it did not resolve the problem. For dsn specify the same value as was provided on the APPLY command.

TRACE

This command activates the IAM/PLEX internal tracing. Refer to the Section 25.41 on IAM/PLEX tracing for additional information on using the trace facility.
IAM/PLEX Tracing Facility

The IAM/PLEX Trace Facility enables Innovation support personnel to request information about I/O requests and other processes that are occurring within the IAM/PLEX Address Space. Trace output will be to dynamically allocated SYSOUT spool datasets or to previously allocated output files. All trace requests are initiated via the z/OS console MODIFY (F) command. If additional parameters are required beyond what was initially specified, then the trace processor will request them via a WTOR. For trace start requests, the additional parameters can alternatively come from a sequential data set.

The IAM/PLEX tracing is only to be used when and as instructed by IAM technical support.

The general format of using the IAM/PLEX trace command is as follows:

```
Fiamplex,TRACE,subcommand,parameters
```

The “iamplex” is replaced by the IAM/PLEX address space name. The subcommands are shown in the table below. The parameters will be as indicated for the subcommand. There are three IAM trace subcommands for IAM/PLEX and all are entered via the z/OS MODIFY command. The subcommands are:

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>Request a new trace</td>
</tr>
<tr>
<td>STOP</td>
<td>Stop an existing trace</td>
</tr>
<tr>
<td>LIST</td>
<td>List all active trace requests</td>
</tr>
</tbody>
</table>

Starting a Trace

To start a trace, issue the command:

```
Fiamplex,TRACE,START
```

The additional parameters can be included on the above command, provided via replies to the WTOR’s, or with the PARM = parameter, through a sequential data set. The parameters for starting a trace are shown on the next page.
### Trace Parameters

Parameters for TRACE START subcommand are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSN=</td>
<td>Optional parameter that specifies the name of the IAM dataset that you wish to trace requests against. Trace information will be generated by any request from any address space that is accessing the named file unless further limited by JOBNAME= and/or STEPNAME= parameters.</td>
</tr>
<tr>
<td>END</td>
<td>Required parameter to signify that all of the desired parameters had been specified.</td>
</tr>
<tr>
<td>ID=</td>
<td>Optional parameter that specifies a unique 1 to 8 alphanumeric character string used to identify this trace request. If one is not specified, IAM will generate a value. This ID is required for STOP and LIST requests.</td>
</tr>
<tr>
<td>INTERNAL</td>
<td>Specifies that trace will be done to an internal wrapping table in 64-bit virtual storage. The table size will be 2 megabytes. This is primarily intended for gathering additional information as may be requested to diagnose a problem.</td>
</tr>
<tr>
<td>JOBNAME=</td>
<td>Optional parameter that specifies tracing is to be performed for requests from specific job. If not specified, all jobs with IAM data sets that are accessed through the IAM/PLEX address space will be eligible for tracing. Use this parameter with DSN= and/or STEPNAME= to limit what is traced.</td>
</tr>
<tr>
<td>JOB=</td>
<td></td>
</tr>
<tr>
<td>OUTDSN=</td>
<td>Optional parameter that causes trace output to be sent to a file instead of a dynamically allocated SYSOUT file. If specified, it must be a pre-allocated and cataloged sequential dataset. IAM will set up the required DCB characteristics.</td>
</tr>
<tr>
<td>PARMS=</td>
<td>Optional parameter that specifies the name of a sequential file that contains the actual trace request. If not specified, all of the necessary parameters must either be specified on the modify (F) command itself, or provided as replies to the WTOR messages that will be issued.</td>
</tr>
<tr>
<td>STEPNAME=</td>
<td>Optional parameter that specifies to trace only requests to IAM/PLEX from the specific step name. When used without other limiting parameters, this will cause ALL requests from any job that has the specified STEPNAME to be traced. When not specified, all step names are eligible for tracing. Use with DSN= and/or JOBNAME= to limit what is traced.</td>
</tr>
<tr>
<td>TRACEREQUEST=</td>
<td>Required parameter that specifies what type of trace records are to be produced. Any or all options may be specified on any trace request. The options are:</td>
</tr>
<tr>
<td></td>
<td>IOS - Trace I/O Start requests</td>
</tr>
<tr>
<td></td>
<td>IOE - Trace I/O End requests</td>
</tr>
<tr>
<td></td>
<td>IOERR - Trace I/O requests that fail</td>
</tr>
<tr>
<td></td>
<td>BFR - Trace IAM Buffer requests</td>
</tr>
<tr>
<td></td>
<td>EXCP - Trace IAM EXCP requests</td>
</tr>
<tr>
<td></td>
<td>XTND - Trace IAM extend requests</td>
</tr>
<tr>
<td></td>
<td>PC - Trace IAM/PLEX PC calls</td>
</tr>
</tbody>
</table>
Stopping a Trace

To terminate tracing, issue the following command:

```
F iamplex,TRACE,STOP,ID=traceid,END
```

The parameters for TRACE STOP requests are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID=</td>
<td>Required parameter that specifies the ID of the TRACE request to be stopped.</td>
</tr>
<tr>
<td>END</td>
<td>Required to signify that all parameters have been specified.</td>
</tr>
</tbody>
</table>

Displaying Active Trace Requests

Use the TRACE,LIST command to display the currently active trace requests. You can request either all of the currently active trace requests, or a specific trace request.

The parameters for TRACE,LIST requests are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID=</td>
<td>Optional parameter that specifies the trace ID from the TRACE,START request. If not specified, all active trace requests will be displayed on the z/OS console.</td>
</tr>
<tr>
<td>END</td>
<td>Required parameter to indicate that all of the desired parameters have been specified.</td>
</tr>
</tbody>
</table>

Examples of Trace Commands

The following demonstrate some examples of using the IAM/PLEX trace function. All examples assume an IAM/PLEX address space job name of IAMRLS.

To trace all I/O Start and I/O End requests from all IAM files from CICS region CICSIDP1:

```
F IAMRLS,TRACE,START,ID=IAM1,JOBNAME=CICSIDP1,TRACEREQ=(IOS,IOE),END
```

To trace all PC calls against dataset MY.IAMFILE:

```
F IAMRLS,TRACE,START,ID=PCTRACE,DSN=MY.IAMFILE,TRACEREQ=PC,END
```

To list all active trace requests:

```
F IAMRLS,TRACE,LIST,END
```

To stop a trace request with the ID of TRACE1:

```
F IAMRLS,TRACE,STOP,ID=TRACE1,END
```
IAM/PLEX CICS Considerations

Overview

CICS TS (Transaction Server) customers can use IAM files under IAM/PLEX, without changes to their application programs or transactions. (Note: IAM/PLEX cannot be used for releases prior to CICS/TS). Additionally there are no changes required to the IAM file definitions under CICS. There are a couple of IAM provided CICS exit routines that must be installed and activated to use IAM/PLEX under CICS. These exits are needed so IAM/PLEX can provide the required record locking protocols. Exits are used for IAM/PLEX to recognize when a logical unit of work (UOW) completes due to a SYNCPOINT request, or transaction termination, which is done using the CICS Task Related User Exit (TRUE) facility. IAM also provides a CICS Global User Exit (GLUE) for backout to recognize when a transaction backout is taking place that is subsequent to the original transaction, such as can occur during an emergency restart. If these exits are not activated, then attempts to OPEN IAM files for IAM/PLEX processing under CICS will be failed.

There is an additional Global User Exit (GLUE) that is provided and must be installed if it is desired that IAM abend any transactions that have encountered lock contention, and have indicated that IAM/PLEX is to ABEND such transactions.

IAM/PLEX CICS Exits

IAM customers that wish to use the IAM/PLEX functionality in their CICS regions must install and activate the provided Task Related User Exit (TRUE), IAMBCICS, the provided Global User Exits (GLUE), IAMXFCBO and IAMXFCRQ, the CICS transaction program IAMXCNTL. To do this, the following steps are required.

- Insure that modules IAMXCINI, IAMBCICS, IAMXFCBO, IAMXFCRQ and IAMXCNTL are in a data set that is part of the DFHRPL concatenation in the CICS JCL.

- Insure that modules IAMXCINI, IAMBCICS, IAMXFCBO, IAMXFCRQ and IAMXCNTL are defined to CICS in DFHCSD. Each of the four modules (IAMXCINI, IAMBCICS, IAMXFCBO and IAMXFCRQ) must be defined with the following attributes:
  - LANGUAGE=ASSEMBLER
  - RELoad=NO
  - DATALOCATION=ANY
  - EXECKEY=CICS

- The module IAMXCNTL must be defined with the following attributes:
  - LANGUAGE=ASSEMBLER
  - RELoad=NO
  - DATALOCATION=ANY
  - API=OPENAPI
  - EXECKEY=CICS

- Alternatively, one can use the CICS batch utility program DFHCSDUP to define the IAM exits. The following JCL example shows how to run the CICS DFHCSDUP program to define the IAM exits to a specified CICS CSD (CICS System Definition file). The input as specified by the SYSIN DD card is the ICL (IAM Installation Control Library) member IAMEXITS which should be reviewed prior to running the update job shown below.
IAM/PLEX and CICS Initialization

Using IAM/PLEX and CICS requires that the IAM exits are defined and enabled during both the stage II and the stage III initialization phases of CICS. The IAM program IAMXCINI performs this initialization of the necessary exits and additional internal storage areas for IAM and must be added to the CICS PLTPI program table. The program IAMXCINI must be added to the second and third stage of the PLTPI list that is used at CICS startup. An example of how this might look is:

```
DFHPLT TYPE=INITIAL,SUFFIX=X1
DFHPLT TYPE=ENTRY,PROGRAM=IAMXCINI For CICS/TS 3.1+
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
DFHPLT TYPE=ENTRY,PROGRAM=IAMXCINI
DFHPLT TYPE=FINAL
END
```
The IAM program IAMXCINI, in the PLTPI functions as an exit activator within CICS to initialize the IAM and CICS environment. It will programmatically issue the following CICS SPI (System Program interface) commands:

- EXEC CICS ENABLE PROGRAM(IAMXFCBO) EXIT(XFCBOUT) START
- EXEC CICS ENABLE PROGRAM(IAMXFCRQ) EXIT(XFCRQ) START
- EXEC CICS ENABLE PROGRAM(IAMXFCRQ) EXIT(XFCRQC) START
- EXEC CICS ENABLE PROGRAM(IAMBCICS) TASKSTART START

These commands activate the CICS customization exits at GLUE and TRUE points respectively for required IAM/PLEX functionality within the CICS region.

During CICS Initialization stages, IAM will issue to the CICS job log the following console messages:

**CICS PLT Stage II messages:**

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFHSI8420I</td>
<td>CICSB319</td>
<td>About to link to PLT programs during the second stage of initialization.</td>
</tr>
<tr>
<td>IAML1003</td>
<td>IAMXCINI - IAM PLT Stage II initialization has begun for IAM v9.2/00A.</td>
<td></td>
</tr>
<tr>
<td>IAML1004</td>
<td>IAMXCINI - IAM TRUE exit program: IAMBCICS has been ENABLED.</td>
<td></td>
</tr>
<tr>
<td>IAML1005</td>
<td>IAMXCINI - IAM GLUE exit program: IAMXFCBO has been ENABLED at EXIT(XFCBOUT).</td>
<td></td>
</tr>
<tr>
<td>IAML1005</td>
<td>IAMXCINI - IAM GLUE exit program: IAMXFCRQ has been ENABLED at EXIT(XFCREQ).</td>
<td></td>
</tr>
<tr>
<td>IAML1005</td>
<td>IAMXCINI - IAM GLUE exit program: IAMXFCRQ has been ENABLED at EXIT(XFCREQC).</td>
<td></td>
</tr>
<tr>
<td>IAML1009</td>
<td>IAMXCINI - IAM Initialization ENQUEUE was issued.</td>
<td></td>
</tr>
<tr>
<td>IAML1006</td>
<td>IAMXCINI - IAM GLUE exit program: IAMXFCBO has been STARTED at EXIT(XFCBOUT).</td>
<td></td>
</tr>
<tr>
<td>IAML1006</td>
<td>IAMXCINI - IAM GLUE exit program: IAMXFCRQ has been STARTED at EXIT(XFCREQ and XFCREQC).</td>
<td></td>
</tr>
<tr>
<td>IAML1003</td>
<td>IAMXCINI - IAM PLT Stage II initialization has ended.</td>
<td></td>
</tr>
<tr>
<td>DFHSI8424I</td>
<td>CICSB319</td>
<td>Control returned from PLT programs during the second stage of initialization.</td>
</tr>
</tbody>
</table>

**CICS PLT Stage III messages:**

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFHSI8430I</td>
<td>CICSB319</td>
<td>About to link to PLT programs during the third stage of initialization.</td>
</tr>
<tr>
<td>IAML1003</td>
<td>IAMXCINI - IAM PLT Stage III initialization has begun for IAM v9.2/00A.</td>
<td></td>
</tr>
<tr>
<td>IAML1011</td>
<td>IAMXCINI - IAM program: IAMXCINI was loaded at x(8B440028).</td>
<td></td>
</tr>
<tr>
<td>IAML1011</td>
<td>IAMXCINI - IAM program: IAMBCICS was loaded at x(800A1000).</td>
<td></td>
</tr>
<tr>
<td>IAML1011</td>
<td>IAMXCINI - IAM program: IAMXFCBO was loaded at x(8B446028).</td>
<td></td>
</tr>
<tr>
<td>IAML1011</td>
<td>IAMXCINI - IAM program: IAMXFCRQ was loaded at x(90E5FA00).</td>
<td></td>
</tr>
<tr>
<td>IAML1011</td>
<td>IAMXCINI - IAM program: IAMXFCAB was loaded at x(8B444BD8).</td>
<td></td>
</tr>
<tr>
<td>IAML1011</td>
<td>IAMXCINI - IAM program: IAMXFCAT was loaded at x(8C00028).</td>
<td></td>
</tr>
<tr>
<td>IAML1012</td>
<td>IAMXCINI - IAM program: IAMXMON was not loaded.</td>
<td></td>
</tr>
<tr>
<td>IAML1011</td>
<td>IAMXCINI - IAM TRUE exit program: IAMBCICS has been STARTED.</td>
<td></td>
</tr>
<tr>
<td>IAML1003</td>
<td>IAMXCINI - IAM PLT Stage III initialization has ended.</td>
<td></td>
</tr>
<tr>
<td>DFHSI8434I</td>
<td>CICSB319</td>
<td>Control returned from PLT programs during the third stage of initialization.</td>
</tr>
</tbody>
</table>
The very first IAM file that is shared via IAM/PLEX will cause at OPEN time, the following console messages to occur:

```
+IAML1000 IAMRLS/CICS Initialization complete. IAM RLS Support active v9.2/00 A (000A1000)
+IAML1001 IAMRLS/CICS is connected to IAM RLS with RLSID = RLS1
```

**CICS Automatic Disconnect and Reconnect after a restart of IAM/PLEX**

Within a CICS region, IAM has the capability to automatically disconnect and then subsequently reconnect to an IAM/PLEX instance after the IAM/PLEX address space has been terminated and then has restarted. This eliminates the need to restart or recycle one or multiple CICS regions that are sharing IAM datasets via the IAM/PLEX option.

The IAM code that does the setting of CICS FILEIDs closed, attempting the RETRY of shunted UOWs and then setting the FILEIDs OPEN and AVAILABLE, runs underneath the default transaction ID of $IAM, unless a different transaction ID has been specified in the IAM Global Options for the CICSTXID= operand. The transaction will run as an CICS OPENAPI subtask so as to facilitate the speedy disconnect and reconnects and closing and opening of IAM files affected by an IAM/PLEX instance being terminated and restarted.

**Disconnect Process**

An IAM/PLEX instance can terminate for a variety of reasons such as an operator issued a normal IAM/PLEX QUIESCE command or an MVS "P" (stop or halt) command to the IAM/PLEX instance or due to an abnormal termination or cancelling of the IAM/PLEX instance. In either situation, IAM code within the CICS region determines that the IAM/PLEX region to which it is connected has terminated. Initially the IAM code in CICS will build and maintain a list of the IAM files that were open ed in CICS and being shared via the affected IAM/PLEX instance. While IAM is building this list of affected IAM files within the CICS region, IAM will attempt to set each of these IAM files to a CICS "CLOSED" state when there are no active UOWs (unit-of-work) for the IAM file. However if there are active UOWs for an affected IAM file, then internally IAM code will issue a CICS "FORCECLOSE" command. This "FORCECLOSE" command will cause any CICS task(s) that have active UOWs to the IAM file to attempt to go through normal CICS backout. That is they will be "rolled back" (any updates in flight will be backed out). However, as CICS will see the IAM file as being in a "not available" status (due to the IAM/PLEX instance having been terminated) CICS will cause the UOWs to be retained, logged, and converted into "shunted UOWs". These "shunted UOWs" will later get resolved when the affected IAM file becomes available again.

**Disconnect Messages**

There are a variety of different messages that may be seen in different places when an IAM/PLEX instance is not longer active. Below are examples of these messages:

On a CICS terminal for inflight transactions at time of disconnect:

```
DFHAC2205 13:00:55 CICSB329 A backout failure has occurred during syncpoint processing for transaction XXXX
```

On the system console and the CICS joblog:

```
IAML1001 IAMRLS/CICS has disconnected from IAM RLS with RLSID = RLB9 (LOCAL)
```
On the system console and CICS joblog when the IAM/PLEX address space is on the same LPAR as the CICS region:

On the system console and joblog when the IAM/PLEX instance is on another LPAR:

```
IAML2005 IAMXCSF1 HAS BEEN POSTED FOR RLSID = RLB9
IAML0174 REMOTE NAME TOKEN AREA DELETED FOR RLSID = RLC9
IAML0173 RLSID=RLC9 IS NO LONGER CONNECTED TO GROUP RLSGRP02
+IAML1001 IAMRLS/CICS has disconnected from IAM RLS with RLSID = RLC9 (REMOTE)
+DFHFC4700 CICSB419 214
A VSAM error has occurred during file backout. (Module DFHFCFR has returned reason code (X'59'), access method code (X'08B8000') and length error code (X'00').)
+DFHME0116 CICSB419 215
(Module:DFHMEME) CICS symptom string for message DFHFC4700 is
PIDS/65551500 LVEL/650 MS/DFHFC4700 RIDS/DFHFCRC PTF/UK37688
PCSS/REASONCODE PRCS/00000059 PCSS/ACCMETHCOD PCSS/E PRCS/08B8000
PCSS/LENGERRCOD PCSS/E PRCS/00000000
+DFHDU0205 CICSB419 A SYSTEM DUMP FOR DUMPCODE: FC4700 , WAS SUPPRESSED BY THE DUMP TABLE OPTION FOR THIS DUMPCODE
+DFHFC4701 CICSB419 217
11/19/2012 18:05:15 CICSB419 Backout failed for transaction CECI, VSAM file KSD50, unit of work X'CA2A87F954E92905', task 07536, base IAMCICSV.$IAM.KSD50, failure code X'FE'.
+DFHDU0205 CICSB419 A SYSTEM DUMP FOR DUMPCODE: FC4701 , WAS SUPPRESSED BY THE DUMP TABLE OPTION FOR THIS DUMPCODE
```
On the CICS Destination Log CSMT (DD=MSGUSR):

<table>
<thead>
<tr>
<th>IAML1050</th>
<th>Processing of CLSF request has begun for RLSID=RLB9</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFHFC0201</td>
<td>11/16/2012 17:13:59 CICSB329 Non-RLS file KSDS0 has been deallocated. Module DFHFCN.</td>
</tr>
<tr>
<td>IAML1051</td>
<td>Processing of CLSF request has completed for RLSID=RLB9</td>
</tr>
</tbody>
</table>

Reconnect Process

When the IAM code within a CICS region determines that the IAM/PLEX address space has been subsequently restarted, IAM will issue a reconnect message and then using the saved list of IAM files IAM will attempt to reopen each of the affected files. Initially IAM will attempt a SET DATASET RETRY for each of the associated IAM datasets for each of the corresponding CICS FILEIDS in the saved IAM file list. This "RETRY" option will cause any existing "SHUNTED UOWs" to be retried. That is the previously failed or shunted units of work that were in flight at the time that the IAM/PLEX address space terminated will be rolled back (backed out). After the "shunted UOWs" have been retried, then IAM causes the affected CICS FILEIDs to be set to an OPEN and ENABLED state. At this point normal processing and access to the IAM files via the connected IAM/PLEX region can proceed.

Reconnect Messages

Shown below are examples of messages that can appear on the console and CICS joblog when IAM/PLEX has been restarted:

<table>
<thead>
<tr>
<th>IAML1001</th>
<th>IAMRLS/CICS has reconnected to IAM RLS with RLSID = RLB9 (LOCAL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+IAML1001</td>
<td>IAMRLS/CICS has reconnected to IAM RLS with RLSID = RLC9 (REMOTE)</td>
</tr>
<tr>
<td>IAML2005</td>
<td>IAMXCLS1 HAS BEEN POSTED FOR RLSID = RLB9</td>
</tr>
<tr>
<td>IAML2008</td>
<td>IXQUERY RC=00000000  RSN= 00000000</td>
</tr>
<tr>
<td>IAML1019S</td>
<td>RLSID= RLC9 CONNECTED TO GROUP RLSCP02</td>
</tr>
<tr>
<td>+IAML1001</td>
<td>IAMRLS/CICS has reconnected to IAM RLS with RLSID = RLC9 (REMOTE)</td>
</tr>
</tbody>
</table>

| DFHFC0970 | CICSB329 Recoverable non-RLS file KSDS0 opened with VSAM SHROPT 3 or 4. CICS cannot ensure integrity. |
| DFHFC0970 | CICSB329 Recoverable non-RLS file KSDS1 opened with VSAM SHROPT 3 or 4. CICS cannot ensure integrity. |

On the CICS Destination Log CSMT (DD=MSGUSR):

<table>
<thead>
<tr>
<th>IAML1050</th>
<th>Processing of RTRY request has begun for RLSID=RLC9</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFHFC0200</td>
<td>11/16/2012 17:15:41 CICSB329 Non-RLS file KSDS1 has been allocated to data set IAMCICSV.$IAM.KSDS1. Module DFHFCN.</td>
</tr>
<tr>
<td>IAML1064</td>
<td>OPEN completed for FILE(KSDS1) with DSN(IAMCICSV.$IAM.KSDS1) at 17:15:37.</td>
</tr>
<tr>
<td>IAML1051</td>
<td>Processing of RTRY request has completed for RLSID=RLC9</td>
</tr>
</tbody>
</table>
For some IAM/PLEX users it may not be practical to deal with the “ILLOGIC” condition raised in CICS when a transaction has timed out while waiting for a record lock. This can occur if a customer has specified that time outs should occur when waiting for locks under IAM/PLEX on the CICSLOCK parameter (See Section 25.10). For those users, a program is provided that runs as a CICS Global User Exit (GLUE), which will purge transactions that receive the IAM/PLEX specific ILLOGIC condition.

A site wishing to activate this functionality needs to do the following:

- Insure that modules IAMXFCAB and IAMXFCAT are in a data set that is part of the DFHRPL concatenation in the CICS JCL.
- Insure that modules IAMXFCAB and IAMXFCAT are defined to CICS in DFHCSD.
- Add IAMXFCAT to the second stage of the PLTPI list that is used at CICS startup. (This is after the DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM.)
- Restart CICS.

The exit activator program, IAMXFCAT, programmatically issues the CICS SPI command:

```
"EXEC CICS ENABLE PROGRAM(IAMXFCAB) EXIT(XFCREQC) START".
```

This command causes CICS to enter IAMXFCAB at the conclusion of each file request made through the standard CICS API before returning control to the application logic.

The exit program IAMXFCAB itself checks the EIBRCODE for the IAM/PLEX deadlock specific indicators, and if they are found it programmatically issues a command to purge the transaction. This results in a transaction abend with a code of AKC3. Users that decide to use this feature will have to determine if they wish to suppress this dump code or use the dumps to identify the files and the activities that are involved in deadlock situations.

Users that have other program(s) enabled at the GLUE exit point “XFCREQC” should generally enable IAMXFCAB last, CICS will run programs at exits in the order they were enabled.
25.60 IAM/PLEX Batch Considerations

Overview

The ability to maintain and achieve data integrity can be a complex matter when dealing with non-transactional processing, as is the case with many batch applications. Such batch applications range from performing a relatively simple and straightforward update process on one or only a few files, to very complex processing involving updates to a large number of related datasets and databases. Additionally, such applications may not have been designed or written with record level data sharing as a consideration. While software products, such as IAM/PLEX, can support and facilitate data sharing, there may be a need to change some batch applications to facilitate good performance and data availability in a data sharing environment.

The discussion of IAM/PLEX and batch processing below is intended to provide assistance in determining what, if any, changes may be necessary to the batch job streams that will be processing files managed by IAM/PLEX. The main considerations that should be kept in mind are what are the recovery requirements for the files being processed by the batch jobs and what other concurrent activity is likely to be occurring on the shared files while the batch job(s) are executing. One of the key points is that with concurrent updating going on, you cannot just restore all of the files that were updated by the batch job and rerun it, because updates made by the other concurrent processing will then be lost. Different recovery techniques will have to be used.

IAM/PLEX and Batch Jobs

IAM/PLEX can be used with batch jobs that read and update IAM files. With IAM/PLEX automatic dataset selection, there are no changes required to JCL for batch jobs to use IAM files under IAM/PLEX. The main area that requires some consideration is for those batch jobs that update recoverable IAM files with IAM/PLEX. For purposes of this discussion recoverable IAM files refers to those files that are utilizing the IAM/PLEX before image journaling. Because such files can have updates backed out should the batch job fail, the record locks are retained until the batch job step ends normally, until a batch syncpoint call or automatic syncpoint is performed, or if the job step fails, until a recovery is performed.

A job step failure from the IAM/PLEX viewpoint is an abend. IAM/PLEX does not look at application return codes, or the condition codes set for a particular job step. If the job step abends, it is considered a failure. If a job step terminates without abending, then IAM/PLEX considers it successful. This is significant because if a recovery is needed, a valid recovery would consist of only recovering to either the last syncpoint, if any, or to the beginning of the job step if no syncpoints were taken. Attempting to backout updates performed by prior job steps may cause a loss of data, because other jobs or online systems may have updated those records subsequent to their completion. Those updates would be lost if a recovery is performed for job steps other than the failing job step. Likewise, attempting to backout updates for a job step the application considers failed due to condition codes could also result in data loss. Backing out of all of the updates for such a job could only be safely performed if the only update access was being performed by the failing job. This can only be guaranteed if the failing job allocates the IAM dataset(s) with DISP=OLD. Otherwise, there is nothing to prevent the IAM files from being updated by other jobs or online systems.

Batch jobs that only read shared IAM files can be executed under IAM/PLEX generally without any JCL or programming changes. Batch job steps that run relatively quickly, say in less than 15 to 30 minutes, and that update a relatively small portion of the records in the dataset, say less than 10%, and that acquire record locks that do not exceed the MAXLOCKS value should be able to use IAM/PLEX recoverable files without any changes.
Batch jobs that update IAM/PLEX files, which have long run times or update a large portion of the records in recoverable files may need to be changed, particularly if they are running concurrently with online systems or other batch jobs that are also updating the same file(s).

IAM/PLEX has a syncpoint capability that can be used by batch applications. When recoverable files are being updated, the record locks are retained for records that are updated, added, or deleted until the job step terminates. If many records are being updated, then the number of locks held could be quite high, resulting in effectively locking out CICS transactions until the batch job completes. Such lock retention could also result in deadlocks between CICS and the batch job, resulting in failed requests. To prevent these types of problems, customers are encouraged to consider the use of the IAM batch syncpoint for such programs. Customers with other VSAM sharing packages may already have such logic built into their batch programs, and will only need to change the syncpoint program they are calling.

The IAM batch syncpoint service will perform a few different functions. First, it will make sure that all updated blocks for IAM files that are being processed by IAM/PLEX for this job have been physically written to DASD. If blocks have not yet been written, such output I/O will be performed. Upon completion of any outstanding buffer writes, the batch syncpoint will then write out a syncpoint record in the journal that can be used if a back out recovery is necessary. Then, any record locks held by that batch job will be released.

The amount of effort to implement a syncpoint within a batch job, along with the associated restart processing may be minimal, or may be rather substantial. This all depends on the nature of the updates being performed, the restart capabilities that the application program may already have, and how other updated files are handled. There are certain types of updates for which an application program can simply be restarted, and it can reapply those updates without the need for any recovery, and retain full data integrity. An example of this could be a job that updates addresses of customers. Such updates may not matter if they were already done before, the program will just end up repeating the update with the same data, or the application may have a way to check to see if the record was previously updated, and if so it would not have to reapply the update. For these types of programs, all that would need to be done is to periodically call the IAM batch syncpoint to release the locks of records updated to that point in time, or use the automatic syncpoint function.

For other applications, such as those that update account balances, a more complex approach may be necessary. These types of programs will need to implement a more complex syncpoint and restart mechanism, such that the program can be restarted from a specified point in time, and be able to resume processing from that syncpoint. When an application is able to do that, it can issue IAM batch syncpoints. If it needs to be restarted, back outs can be performed to for example the most recent syncpoint and the job will not perform any double updates. The batch application program will in effect, have to perform a processing checkpoint, and include in that the execution of the IAM batch syncpoint. The application will have to implement their own type of checkpoint processing, as z/OS Checkpoint cannot be used with IAM files being processed under IAM/PLEX.

Calling the IAM Batch syncpoint is easy. Just issue a call to IAMSYNC. No parameters are necessary. The program IAMSYNC is a small stub, and designed to be link-edited in with the application program. An alternative is for the application to LOAD the IAMSYNC module, saving the address and calling it as necessary.
As an alternative to revising the actual program, IAM/PLEX has an automatic syncpoint capability. This ability is enabled by use of the IAM ACCESS override of AUTOSYNCPOINT=nnnnn. This override is specified on the job step that is updating IAM files through IAM/PLEX for a selected key file that is processed using IAM/PLEX. The AUTOSYNCPOINT specifies that after the specified number of I/O requests that result in record lock being acquired are performed, IAM will automatically invoke the batch syncpoint process. This facility is useful for circumstances where it is not possible to change the application programs.
Recovery Overview

One of the most important considerations when implementing IAM/PLEX, or any type of data sharing, is how to recover should errors, media failures, or outages occur. The journaling and recovery tools provided with IAM will assist in the recovery of data for the IAM files that are processed under IAM/PLEX. The intent of the information presented in this section is to assist you in planning for your data recovery needs, and to help insure that appropriate procedures are put in place for the protection of your data.

This section will address performing backout recoveries for failing batch jobs, and forward recoveries from restored IAM datasets. CICS provides the recovery for failed transactions with dynamic transaction backouts, so IAM/PLEX does not provide that capability.

IAM/PLEX Journals

In order to perform recoveries with the IAM tools, you must provide for IAM/PLEX Journaling, as described in Section 25.30. You'll need to decide on what type of recoveries you'll want to be able to perform. If you want to be able to backout updates from failing job steps, you'll need to specify the journaling of before images. If you want to be able to recover files after restoring them from a backup, you'll need to be collecting the after images. To provide the best possible recoverability, Innovation recommends that you journal both before and after images. This way, you'll be able to perform whatever recoveries may be necessary. To insure that journaling is being performed for all of your files, it is recommended that you provide the appropriate ACCESS overrides to the IAM/PLEX address space that will specify the level of journaling you need. For example, to specify journaling for all datasets, you can use the following IAM override control card for the IAM/PLEX proc:

```
ACCESS DD=&ALLDD,JRNAD=BOTH
```

If you have other override cards, be sure to specify the JRNAD=BOTH on those as well, as the DD=&ALLDD override card is effective only for datasets that are not otherwise explicitly specified on override control cards.

Dynamic Job Backout

IAM/PLEX offers an optional capability to dynamically backout changes made to IAM/PLEX files that were made by an abending job step. The use of Dynamic Job Backout is controlled by the IAM/PLEX parameter DJB, as discussed in Section 25.10. When a job step that has updated IAM files managed by IAM/PLEX abends, if Dynamic Job Backout can be used by that job, then IAM/PLEX will internally schedule the backout. The backout will be performed under the IAM/PLEX instance that the job was directly connected to. Appropriate messages issued upon completion. Upon completion of the backout, IAM/PLEX will release any record locks retained by the failing job step. If the abending job step had taken any IAM syncpoints then the backout will be performed to the last syncpoint taken by the job step. If no syncpoints are found, then all updates made to IAM files managed by IAM/PLEX will be backed out.
Manual Job Backout

If you have not enabled Dynamic Job Backout, and a batch job step abends, you can use IAMJREST to perform a backout of the failing job step. You would not normally want to perform a backout of the updates made by the preceding job steps because those record locks would have been released upon normal step termination, and those records could have subsequently been updated by another batch job or online system. An example of a manual job backout is shown below. IAMJREST will automatically release any retained locks upon completion of the backout.

```plaintext
//BACKOUT EXEC PGM=IAMJREST,REGION=0M
//SYSPRINT DD SYSOUT=*  
//SYSOUT DD SYSOUT=*  
//IAMINFO DD SYSOUT=*   
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50,10))  
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50,10))  
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50,10))  
//SYSIN DD *  
  RESTORE BACKOUT,JOBNAME=jobname,STEPNAME=stepname,  
  JOBID=JOBnnnnn  
/*
```

Figure 25-13: Example of JCL for manual backout (EX2570C)
A forward recovery may be necessary if an IAM file is damaged due to hardware media failures, or due to software failures. The first step for such a recovery is to restore the IAM dataset from a backup copy. The backup could have been done by an FDRREORG, IDCAMS, or some other application, or it could have been done by a DASD management utility such as FDR or DFSMSdss. Once the dataset is restored, you can use IAMJUTIL to reapply all of the updates that had occurred to the dataset prior to the failure. For the forward recovery, you would want to specify the FROMDATE and FROMTIME being the time that the backup started, and specify the IAM dataset that is being recovered.

In the example below, there is the presumption that the current IAM/PLEX journal log stream has all of the necessary data to perform the recovery, so no journal datasets are specified in the JCL.

```sql
//RECOVER EXEC PGM=IAMJREST,REGION=0M
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//IAMINFO DD SYSOUT=* 
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50,10)) 
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50,10)) 
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50,10)) 
//SYSIN DD " 
  RESTORE FORWARD,DSN=bad.iam.file,FROMDATE=yyyyddd, FROMTIME=hhmmss 
/*
```

*Figure 25-14: Example of a Forward Recovery (EX2070D)*
30 IAM 9.2 OVERRIDES

IAM Override Statements Overview

IAM offers various features and capabilities that are not available with VSAM. One of the ways to indicate to IAM what features and capabilities are to be used with any specific file is through the IAM Override facility. Many features can be activated or deactivated as defaults through the IAM Global Options facility. Others, such as IAM's Dynamic Tabling, are available only through an override specification. There could also be circumstances where the user does not want the default values from the IAM Global Options Table, or wants to make sure a certain capability is being used. For example, with a very highly accessed IAM file, the user may want to increase the maximum and/or minimum number of buffers that IAM is to use for processing that file.

This section is intended to provide in one place, a reference for all of the IAM Overrides that are applicable with IAM Version 9.2 using the IAM VSAM (VIF) interface. The various facilities and capabilities of IAM are described in other places of the manual.

IAM offers the user two types of override statements, the CREATE override and the ACCESS override.

CREATE Statement

The CREATE statement can be used when a file is defined, loaded, reorganized or re-cataloged to specify file attributes and special IAM features. Using the CREATE statement, an IAM user can tailor file processing by changing or overriding file characteristics and options that would otherwise be based on the IDCAMS DEFINE parameters and the defaults in IAM's Global Option table. In some cases IAM's default values may not be sufficient. For example, for a file that has very heavy access, which could benefit from using more than the default number of buffers use the CREATE statement to override the defaults for MAXBUFNO and MINBUFNO on the file define or load, which will eliminate the need to override the file by every job that accesses that file.

ACCESS Statement

The ACCESS statement can be used when a loaded file is processed for input or update to override IAM run time options for a specific job. Using the ACCESS statement an IAM user can, at execution time, tailor a file's processing. The user can specify options that do not exist in VSAM (ex: Dynamic Tabling of Data Records) or adjust IAM's Real Time Tuning (ex: MAXBUFNO, MINBUFNO). The overrides specified on an ACCESS statement are used only for that particular job step, and are not saved within the file as certain CREATE override values are.

IAMOVRID DD Card

The IAMOVRID DD card specifies the Override control data set. This data set is in card image format, and is typically an input stream (i.e. DD *) data set or a member of a card image PDS. IAMOVRID may be used in any job step that processes IAM files (IDCAMS, CICS, batch applications, etc.). In general the IAMOVRID file is only read once per job step, unless the REREAD keyword is specified. The file is read when the first IAM file in the job step is opened. A table of the overrides is built in extended private storage, and is referenced for subsequent IAM file opens. The table remains in storage until step termination. Up to two hundred override statements can be provided.

For long running address spaces including CICS, IAM/RLS or IAM/PLEX, IAM will automatically read the overrides for each IAM data set being opened. This enables users to revise the override statements and have the changes take effect without recycling the CICS, IAM/RLS, or IAM/PLEX address space.
IAM Override statements are contained on one or more 80-character records coded in positions 1 through 71. Each control statement starts with a command (ex: CREATE or ACCESS). The command can begin in the first column, or can be preceded by some blanks. The command is immediately followed by one or more blanks, then one or more optional keywords with their values, separated by commas. The last keyword or keyword / value pair in a statement must be followed by one or more blanks. Anything following the blank will be treated as comments. Comments are not permitted prior to the first keyword in a statement. The entire line will be treated as a comment when a '*' is placed in the first column.

To continue a statement onto an additional line a comma and at least one blank must follow the last keyword or keyword / value pair on that line. The next keyword is then begun on the following line. When a keyword requires an optional variable, that keyword and its variable must start and end on the same line.

Each override statement indicates the file(s) to which it applies through the DD= or DSN= keyword. Up to 40 different DD names or one data set name (DSN) can be specified for any particular override statement. Additionally, an override command can be made applicable to all IAM files in that job step by coding DD=&ALLDD. Such an override will be effective for all IAM files except those that have their own explicit override. If multiple overrides of the same type (i.e., CREATE or ACCESS) are specified for the same DD name, then the overrides on the last one will be used for that execution, with ALL values from prior cards being replaced. The DD name field has precedence over the data set name (DSN). The file(s) and data sets referenced on the overrides do NOT have to be included in the JCL. IAM data sets can be dynamically allocated. This is frequently the case with CICS systems, and when using IAM Alternate Index support. When an IAM file is opened, IAM scans through the table of overrides to see if there is a match for the DD name of the file being opened. If so, the specified overrides are used. If no matching DD name is found, then IAM scans through the DSN= overrides looking for a match. If none is found, and the &ALLDD override has not been specified, then the file will be processed using defaults from the IAM Global Options Table.

The various override keywords can be abbreviated, if desired. The underscored portion of the keyword indicates the minimum abbreviation allowed when it is described.

If an error is detected on one or more of the provided override cards, IAM will print the contents of the override card on the job log, along with a message indicating the error. If an IAM files is being defined or loaded and an override error is encountered, then the file definition or load will be failed. Override errors are tolerated for normal file access, with the bad override statement(s) being ignored. The reason a define or file load is failed when there are override errors is that the file may not be defined or loaded with the desired attributes, which may cause processing problems in subsequent programs that use the file. Normal file access overrides, while perhaps critical for performance, will generally not have the impact on a file that the CREATE overrides can have, so therefore those errors are tolerated.
30.02 IAM CREATE Override Statement Format

The CREATE statement can be used to override file characteristics that are stored in a file and processing options that take effect when a file is defined or loaded. This override statement can be used either on the file definition or on the file load step. When used on a file load, the overridden file attributes will have precedence over attributes from the file definition.

Keywords that are related to file load, including BACKUPCOMPRESSED, CRBUFOPT, DATASPACE and TRACEDDNAME are only relevant to an actual file load. Conversely, the UNIT keyword is only relevant during a DEFINE. Note that the overrides will not be referenced for files being defined through JCL allocation. If overrides are desired for such files, they must be specified on the job step that is actually loading the file.

The CREATE override can be used when executing a DEFINE RECATALOG to alter selected file attributes. The values that can be changed on a DEFINE RECATALOG via the CREATE override include BUFSPACE, JRNAD, MAXBUFNO, MINBUFNO, RLSID, and SHAREOPTION.

Selected old CREATE Override keywords are no longer documented in this manual. Information can be found on these keywords in the ICL library member OLDOVRID, which was downloaded as part of the installation procedure.

CREATE DDNAME=ddname  | DSN=dataset name
[,BACKUPCOMPRESSED] [,BLKSIZE=nnnnn]
[,BUFSIZE=nnnnnnnn]
[,CRBUFOPT=64BIT | MCYL | CYL | MTRK | TRK | BSAM]
[,DATACOMPRESS=YES | NO | HW | HARDWARE | SW | SOFTWARE]
[,DATASPACE=nnnn] [,DICTIONARY=cccc]
[,DSNTYPE=LARGE | BASIC | EXT] [,DYNALLOC]
[,ENHANCED] [,HPF=YES | NO]
[,INDEXCOMPRESS=YES | NO | LONG]
[,INTEGRATED=nn] [,JRNAD=BOTH | ALL | BEFORE | AFTER | NONE]
[,LIMITKEYS=nn] [,LOG=YES | NO]
[,MAXBUFNO=nnnn] [,MAXSECONDARY=nnn]
[,MINBUFNO=nnnn] [,MULTIVOLUME= PRIMARY | SECONDARY]
[,OVERFLOW=nnnnnnnn] [,PE=nnnnn]
[,PRO=YES | NO] [,PSEUDORBA]
[,RELEASE=YES | NO] [,REREAD]
[,RLSID=cccc] [,RRDSLOAD= RANDOM | SEQUENTIAL]
[,SHAREOPTION= 1 | 2 | 3 | 4] [,TRACEDDNAME=ddname]
[,UNIT=unitname] [,VAROVERFLOW=YES | NO]
[WKDDNAME=ddname] [,XESDS | NOXESDS]

Figure 30-1: CREATE Override Statement
**DDNAME=**

Specifies the DD name of the IAM data set that the override is to be applied to. Up to 40 DD names may be specified. If multiple DD names are specified they must be enclosed in parenthesis and separated by commas.

**DD=ALLDD** will result in this override being applied as a global override to all IAM files being defined, loaded or re-cataloged in this step, unless otherwise explicitly overridden.

When used during an IDCAMS DEFINE, it is recommended that DD=ALLDD or the DSN= keyword be used. If using the DD= override for a define process, the value MUST match the value specified for the FILE (ddname) keyword on the IDCAMS DEFINE control statement. In this circumstance, there is no need for the actual DD statement as specified by FILE (ddname) or DDNAME= on the CREATE override to be present in the JCL.

An * as the last character of the specified DDNAME indicates that only the characters up to but not including the * should be used to match the actual DDNAMEs used in this step. This wildcard allows a single override to be applied to more than one DDNAME, based on its name. Do not specify DD=* to reference all DDNAMEs in the job. Use DD=ALLDD to do this.

**There is no default. Either DDNAME or DSN must be specified.**

**DSN=**

Specifies the data set name of the IAM data set that the override is to be applied to. One data set name per override control statement is permitted.

An * as the last character of the specified DSN indicates that only the characters up to but not including the * should be used to match the actual data set names used in this step. This wildcard allows a single override to be applied to more than one DSN, based on its name.

**There is no default. Either DDNAME or DSN must be specified.**

**BACKUPCOMPRESSED**

Specifies that the input data is in IAM Data Compressed format. The Backup Compressed option is only valid for IAM software compressed datasets. It is ignored for hardware compressed datasets. It can only be specified on a file load step with data that was created with the IAM ACCESS Override of BACKUPCOMPRESSED. Specification of this option forces the output file to have a data compressed format.

Default is that data is treated as standard, uncompressed data.

**BLKSIZE=**

Specifies a blocking factor or an explicit block size for the IAM file. For values from 1 to 15, IAM will use a block size that will result in the specified number of blocks per track for the device type on which the data set is being allocated to. For values of 300 or larger, IAM will use that value as an explicit block size. Values from 16 to 299 are invalid.

Default is generally 4 blocks per track, based on the IAM Global Option VSAMBLOCKF, the default blocking factor for IAM files defined under the VSAM interface. A larger block size may be used however if the DEFINE indicates a larger value for a Control Interval Size, or fewer than 5 records will fit in the default block size.
**BUFSPACE=**

Specifies the maximum amount of storage that can be used for buffers in kilobytes. If specified other than 0, this value will supersede any specification for MAXBUFNO. This value is saved with the dataset characteristics and will be used to compute MAXBUFNO unless it is otherwise overridden.

MAXBUFNO is computed as (BUFSPACE*1024)/BLOCKSIZE.

Default is buffer space will be determined when the file is being accessed.

**CRBUFOPT=**

Specifies the buffering option to be used for the file load. Because the file load is essentially a sequential output process, the buffering is different than the normal IAM Real Time Tuning Buffering. The MINBUFNO and MAXBUFNO values are not used during the file load. During the file load, IAM acquires a number of buffers based on this parameter. When half of the buffers are filled with data they are written out to the device in a single I/O operation. When the second group of buffers is filled, IAM will wait for the prior I/O to complete, then start the I/O for the second group, and let the application begin filling up the first group of buffers. The buffers are requested from extended private storage. This keyword value is only utilized for a file load step; it is ignored for define. The values that can be specified are:

**BSAM** - Use the BSAM Access Method macros to load the file. The default number of buffers is 12, unless overridden by the BUFNO override. These buffers are all acquired in below the line storage.

**CYL** - Buffers for one cylinder's worth of blocks is acquired, approximately 1/2 cylinder is written per I/O.

**MCYL** - Buffers for two cylinder's worth of blocks is acquired, one cylinder is written per I/O.

**MTRK** - Buffers for ten tracks are acquired, five are written per I/O.

**TRK** - Buffers for two tracks are acquired, one track is written per I/O.

**64BIT** - Same buffering as MCYL except with 64-bit virtual storage used for the buffers.

Default value is MCYL, or as otherwise specified in the IAM Global Options Table.
**DATACOMPRESS=**

Specifies whether or not IAM is to attempt to compress the data content of each record. Valid values are YES, NO, HW, HARDWARE, SW, or SOFTWARE.

Specifying **YES** will enable the use of IAM data compression. The type of data compression used, software or hardware, will be set based on the IAM Global Options Table, which defaults to Software. This will usually result in significant savings of DASD space for the file, reduce virtual storage requirements to process the file, and reduce physical I/O (EXCP's) for the file. Files must have at least 10 bytes of data after the key to be eligible for data compression. If the IAM data compression does not reduce the size of a record, it is placed in the file uncompressed.

Specifying **NO** will result in all of the data records being stored in the file without IAM performing any data compression.

Specifying **HW** or HARDWARE will use the hardware compression instruction. This type of compression requires a compression dictionary. IAM will either dynamically build one during file load, or the user can create a compression dictionary and specify the name using the DICTIONARY override. In either case, the dictionary will be stored within the loaded dataset. Customers may be able to achieve greater compression through the use of the hardware compression facility, particularly if they create a compression dictionary from the data within the file. Hardware compression should not be used on processors prior to the IBM z/Series due to the significant CPU overhead on the older processors. The hardware compression overhead has been substantially reduced on the IBM z/Series processors, and may in some circumstances use less CPU time than the IAM software compression.

Specifying **SW** or SOFTWARE will cause IAM to use its internal software compression technique that does not require a dictionary.

Default is files that are defined with a primary space of 75 tracks (5 cylinders) or larger will automatically be eligible for IAM data compression, and will default to software. This file size and compression technique can be changed in the IAM Global Options Table. The size limitation is ignored when the override is specified.

**DATASPACE=**

Specifies the size, in megabytes, of the Data Space to be used for the temporary storage of the index to the IAM file that is being loaded. This override is ignored during file definition. Valid values are from 0 to 2048. A value of 0 results in the use of a dynamically allocated temporary data set, or other dataset that may be specified via the WKDDNAME override and DD card.

Default value is 2048 megabytes, or as otherwise specified in the IAM Global Options Table.
**DICTIONARY=** Specifies the four-character suffix for the name of the user provided hardware compression dictionary. The dictionary must be in load module format, with the first four characters being ‘IAMD’, and the second four characters will be the customer’s choice. Review Section 10.25 on Hardware Compression for information on creating and naming the compression dictionaries.

Default is to use a dictionary dynamically created by IAM during the file load when hardware compression has been requested.

**DSNTYPE=** Specifies the external IAM file format on DASD, and is only valid for customers using z/OS 1.7 or above.

- **LARGE** indicates that the IAM file should be an MVS Large Format Sequential Dataset.
- **EXT** indicates that the IAM file should be a DFSMS Extended Format file.
- **BASIC** indicates that the IAM file should be a standard PS dataset.

Default is BASIC, unless specified differently by the DFSMS Data Class.

**DYNALLOC** For use when the IAM Global Option of ENABLE=VAM is set to force dynamic allocation instead of the DADSM SVC allocation that is used with the CA Allocate (formerly VAM) product.

**ENHANCED** Specifies that IAM is to create this file in the Enhanced format. Enhanced format KSDS files offer various processing enhancements including the ability to dynamically acquire additional DASD space for updates and inserts that occur after the file is loaded. The use of Enhanced format files is required for IAM ESDS and RRDS files, for IAM files that have an alternate index, or for IAM files that will be used with the IAM RLS facility.

Default is Enhanced format file, unless otherwise specified in the IAM Global Options Table.

**HPF=** Specifies whether or not IAM will use z/HPF architecture on devices that are capable of utilizing that function. Specifying **YES** will cause IAM to use z/HPF architecture on the volume(s) that the system indicates are capable of supporting EXCP processing with z/HPF commands. Specifying **NO** indicates that IAM will not use the z/HPF architecture I/O commands.

Default is YES that z/HPF will be used unless otherwise specified in the IAM Global Options Table.
INDEXCOMPRESS=

Specifies whether or not the file is eligible for a compressed index. **Valid values are YES, LONG or NO.**

When **YES** is specified, the file still has to meet the minimum size requirement as indicated by Global Options which defaults to 8000 bytes. The key length must be between 4 and 128, and the index compression must yield storage savings of at least 10%.

When **LONG** is specified, IAM will use a Long Key Compression algorithm, which does not limit a compression set size to 256 bytes. This will provide index compression for files with keys longer than 128 bytes, and may also provide other files with keys of 32 bytes or longer with better compression.

When **NO** is specified, a compressed index will not be created for the file. For files with high random I/O activity, or with a very large number of records being randomly inserted into the file, not using the compressed index may reduce CPU time, at the expense of potentially substantial increased virtual storage requirements.

Default value is YES except for files with keys of 33 bytes or more which will default to LONG.

INTEGRATED=

Specifies the amount of space, as a percentage, that is to be reserved for future growth within each prime data block during file load. This field is identical to the VSAM CI free space parameter. This value is also used for prime extension blocks as records are being added to the logical end of the file. **Valid values are from 0 to 99.** For most files the use of a large value for CI free space is counter-productive in that it results in a lot of unused space within the file.

Default, for KSDS files, is from the CI FREESPACE value specified during the file define. For ESDS and RRDS files, the default is 0 or as otherwise specified in the IAM Global Options Table for ESDSINTEGRATED or RRDSC1%.
Indicates that the type of journaling, if any, that IAM will perform. When specified on the CREATE override, either during file definition or file load, the journaling specification will be used for any file update operation. Journaling can also be overridden by an IAM ACCESS override statement.

For files not being processed by IAM/RLS or IAM/PLEX, users must pre-allocate and catalog a log data set to be used for the journaling, which is required to be the name of the IAM data set / cluster, appended with the characters ".LOG". IAM journaling will not be active during file loads, reorganizations, or during recovery from the journal.

For files being processed by IAM/RLS or IAM/PLEX, the users must specify the journal data sets to the IAM/RLS or IAM/PLEX startup procedure.

One of the following values can be specified:

**BOTH** - The IAM log data set will contain both before and after images. This will enable the user to perform either a forward recovery, or a backward (backout) type of recovery.

**ALL** - This is a synonym for BOTH. The IAM log data set will contain both before and after images. This will enable the user to perform either a forward or backward (backout) type of recovery.

**BEFORE** - The IAM log data set will contain only before images of updated records. This option allows backwards (backout) recoveries only.

**AFTER** - The IAM log data set will contain only the after images of updated records. This option allows forward recoveries only.

**NONE** - The IAM journaling feature will not be used for this IAM data set.

Default value is NONE.

Specifies a value between 3 and 64, which will indicate the number of keys that will be in a set of compressed keys. The total length of the keys within a set will be limited to the smaller of this value, or the number of keys that will fit within 256 bytes. If long key compression is being used, the 256 byte size of a set of keys is ignored. Varying this number may yield better key compression, resulting in a smaller index, for some files. General recommendation is to use the default value.

Default value is 32.

Indicates whether or not the IAM Override processor is to display this override card on the job log. Valid values are YES or NO. This is not recommended for use with the REREAD option, particularly on CICS regions or for IAMRLS, as it can result in significant output to the job log and system log.

Default value is NO; the override is not displayed unless it is in error.
**MAXBUFNO=**

Specifies the maximum number of buffers IAM will acquire during file processing. IAM's Real Time Tuning dynamically adjusts the number of buffers used for a file as I/O demands change, up to the limit specified or defaults to for MAXBUFNO. While this has no effect during the file load, the value is saved within the file, and will be used as a default value for the file when opened for input or update processing. Valid values are from 0 to 24576.

A value of 0 indicates no override value for MAXBUFNO is set. This can be used with a DEFINE RECATALOG to nullify any previous setting of MAXBUFNO.

Default is 0, that no maximum number of buffers will be associated with the file.

**MAXSECONDARY=**

Specifies a multiplication factor to be used by the IAM Dynamic Secondary Space Adjustment feature. Once a file has five or more extents on a volume, the secondary space value will be increased by the MAXSECONDARY factor, unless it exceeds the primary space value or the amount of space available on the volume. Valid values are from 0 to 10. A value of 0 will disable this feature. MAXSECONDARY is ignored on DFSMS Extended Format files due to large number of extents that they can take per volume.

Default value is 10 during file load, unless otherwise specified in the IAM Global Options table.

**MINBUFNO=**

Specifies the minimum number of buffers that IAM will maintain during file processing. IAM's Real Time Tuning dynamically adjusts the number of buffers used for a file as I/O demands change, but will never use less than the specified or defaulted value for MINBUFNO. Valid values are from 0 to 24576. Specifying an equivalent number of buffers for MAXBUFNO will effectively disable IAM's Real Time Tuning algorithm. If MINBUFNO is overridden by a value greater than 0, then that value will be used as the initial number of buffers to acquire during Open processing. If the value for MINBUFNO is set larger than MAXBUFNO, it will be set down to the MAXBUFNO value.

A value of 0 indicates no override value for MINBUFNO is set. This can be used with a DEFINE RECATALOG to nullify any previous setting of MINBUFNO.

Default is 1 as a minimum number of buffers will be associated with the file.

**MULTIVOLUME=**

Indicates whether IAM is to use the PRIMARY or SECONDARY space allocation value, when IAM anticipates that the next extent will be obtained on another DASD volume. Valid values are PRIMARY or SECONDARY. MULTIVOLUME is ignored for DFSMS Extended Format datasets due to the large number of extents they can take on each volume.

Default value is PRIMARY, based on the IAM Global Options Table.

**NOXESDS**

When specified for DEFINE or LOAD of an ESDS type of IAM file will result in an ESDS file with a standard 4-byte RBA. This is intended to be used to override defaults from the Global Options that would have the ESDS be an Extended Addressable format with an 8-byte RBA or with the IAM PSEUDORBA value. This will also override the Extended Addressable specification from the SMS Data Class.
OVERFLOW= This value will be used to calculate the target amount of DASD space to reserve for future expansion at the end of the file load. This will not guarantee that the overflow area can actually hold this quantity of records. **DASD space will only be reserved within the confines of the currently allocated space.** No additional extents will be taken to meet this target during the file load. When a file reaches or exceeds the number of specified overflow records an IAMW22 message will be displayed on the job log during either open or close processing to indicate that file reorganization is recommended.

Default value is based on CA Free space value and the amount of DASD space actually used for the file at completion of the load.

PE= Specification of PE=0 will circumvent normal VSAM rules by allowing a file to be loaded with 0 records. Standard VSAM requires at least one record to be inserted into a file before it is considered to be in the loaded state.

PRO= Indicates whether or not IAM should use the Prime Related Overflow structure. This format is only recommended for use on datasets that are subject to a large number of inserted records and are not shared for read only while being updated unless it is used under IAM/RLS or IAM/PLEX. Valid values are **YES** or **NO**. PRO files cannot be used with IAM Version 8.1 and earlier versions of IAM.

The default is based on the value in the IAM Global Options Table which is shipped as NO.

PSEUDORBA Specifies that an IAM ESDS file can exceed 4 gigabytes in size, through the use of non-standard 4-byte RBA’s (Relative Byte Addresses). If the application has dependencies on the RBA to be identical to VSAM, then this parameter must not be used. This is helpful for applications that exceed the 4-gigabyte data size, and would require changes to support an 8-byte RBA value.

Default is standard VSAM 4-byte RBA.

RELEASE= Specifies whether IAM is to release unused space after the file is loaded, or reorganized. By default, IAM will release unused space in a file’s allocation after the initial load, providing that secondary space was specified. IAM will reserve some space, if any is available, of the over allocated space based on CA free space, or the Overflow override if it is specified. Valid values are:

**YES** - IAM will release unused, unreserved space after each load or reorganization.

**NO** - IAM will not release unused DASD space. This may be useful to hold on to DASD space for future expansion.

Default is to release unused, unreserved DASD space on the first load only, and only if a secondary space quantity has been specified. The default can be changed in the IAM Global Options Table.
**REREAD**

Indicates that IAM will reread the overrides each time an IAM file is opened. This will be automatically done for CICS, IAM/RLS, and IAM/PLEX address spaces. If specified, this keyword should appear on all of the override cards supplied in the file. This is primarily intended to be used by long running jobs, where the IAMOVRID points to a sequential data set or PDS member. This provides a mechanism for users to change overrides, and have them take effect by closing and reopening the IAM file(s).

Default is that the IAM override parameters are once per job step except and kept in storage except for CICS, IAM/RLS, and IAM/PLEX.

**RLSID[=cccc]**

When specified with no operand on define or load, indicates that this dataset is to be processed using IAM/RLS or IAM/PLEX, regardless of Share Options or the IAM RLS Dataset Name Tables. The operand, if specified, is the four character identifier of the IAM/RLS or IAM/PLEX address space that this dataset is to be processed by.

When specified with no operand on a Define Recatalog process it will nullify any prior RLSID specification.

**RRDSLOAD=**

When loading an IAM RRDS file, specifies whether the RRDS is being loaded in ascending relative record number sequence, or in a random relative record number sequence. The values that can be specified are:

- **SEQUENTIAL** – Specifies that the RRDS file is being loaded in ascending relative record number sequence. This is the default value for variable length record RRDS files. Use this override when loading a fixed length record RRDS file which is being loaded in ascending relative record number sequence when the program indicates random processing.

- **RANDOM** – Specifies that the RRDS file is being loaded in a random relative record number order. This is the default value for fixed length RRDS files when the program indicates random processing. Specify this override to load a variable length record RRDS file in random relative record number sequence, which is a function not provided by VSAM. IAM will use data space(s) to hold the records written until the file is closed. IAM will then sort the records within the data space(s), and write the records to the file in ascending record number sequence.

Default is RANDOM for fixed length record RRDS files, and SEQUENTIAL for variable length record RRDS files.

**SHAREOPTION=**

This value is only used for a Define Recatalog process. It provides a mechanism for changing the share option to a value different than what was specified when it was originally defined or last recataloged. Refer to IDCAMS in Section 10.20 for a description of the various Share Options. Valid values are 1, 2, 3, or 4.

**NOTE:** **SHARING IAM FILES FOR UPDATE IS STRONGLY DISCOURAGED, WITHOUT THE USE OF IAM/RLS, IAM/PLEX OR OTHER SOFTWARE THAT FACILITATES SHARING! SUCH SHARING MAY ADVERSELY AFFECT THE DATA INTEGRITY OF THE FILE WITHOUT THE PROPER SOFTWARE PROTECTION!**

Default is that the file will retain the current Share Option.
**TRACEDDNAME=**
Activates the IAM request trace capability, and indicates where the trace data is to be written to. Normally, it is written to SYSOUT; however it can also be written to a disk or tape file. To generate a printed report, specify a DD name of IAMTRPRx, where 'x' is any valid character. If 'x' is a 'C', then the key of each record is printed in character format, otherwise the key is printed in hexadecimal format.

**WARNING:** The trace to a particular output trace data set can only be done for a single IAM file at a time. Please **DO NOT SPECIFY TRACEDDNAME WITH DD=&ALLDD**. This can result in abends if multiple concurrent traces for different IAM files to the same TRACEDD are being performed.

Default is no trace output is produced.

**UNIT=**
For a file define request, specifies a generic unit name for non-specific volume allocation. Disk volume(s) associated with the specified generic name must be mounted as STORAGE, and the VOLUMES parameter must specify 'ANYVOL' to activate IAM's non-specific volume allocation. This feature cannot be used for DFSMS managed data sets, or for CA-Vantage Allocation Manager, formerly Sterling's SAMS (VAM) managed data sets.

Default is SYSDA, unless otherwise specified by the WORKUNIT field in the IAM Global Options Table.

**VAROVERFLOW=**
Specifies whether or not IAM is to enable variable length record support for the overflow area. **Once a file has variable overflow enabled, it will not be disabled without doing a reorganization.** Valid values are YES or NO. Use of variable length overflow will for many files reduce the amount of DASD space used for the overflow area.

Default is YES, unless specified differently in the IAM Global Options Table.

**WKDDNAME=**
Specifies the DD name of the DD statement specifying a temporary sequential data set used as temporary storage for the file's index as it is being loaded. This is generally not needed, as IAM normally uses either a data space, or its own dynamically allocated temporary data set. However, for cases where the data space usage is not allowed, or cannot be large enough, and the default space allocation of the temporary work data set is insufficient, a user can provide their own data set allocation for this purpose.

Default if using a work data set is IAMWKDD.

**XESDS**
For ESDS type of files, enables the use of 8 byte RBA values. This will allow ESDS files to exceed the 4-gigabyte file size with compatibility to DFSMS. If the Extended Addressability option has been set in the DFSMS Data Class for this ESDS file, then 8-byte RBA support is automatic. This option must not be specified if PSEUDORBA has been specified.

Default value is based on the Extended Addressability from the SMS Data Class or from the IAM Global Options. If neither is set then will default to a standard 4 byte RBA value.
IAM ACCESS Override Statement Format

The ACCESS statement is used to override IAM execution time defaults, for a specific step. The ACCESS override statement applies to file access processing only, it does not apply to file define, load, or create.

**ACCESS Statement Operands**

<table>
<thead>
<tr>
<th>OPERANDS</th>
<th>DDNAME=ddname</th>
<th>DSN=dataset name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[.,ALTLOGGER]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[.,AUTOSYNCP=nnnn]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[.,BACKUPCOMPRESSED]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[.,CACHE64]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[.,DEFERWRITE= YES</td>
<td>NO]</td>
<td></td>
</tr>
<tr>
<td>[.,DYNCORE=nnnn]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[.,ESDSLOCK= YES</td>
<td>NO]</td>
<td></td>
</tr>
<tr>
<td>[.,INDEXSPACE=64BIT</td>
<td>YES</td>
<td>NO]</td>
</tr>
<tr>
<td>[.,LOG = YES</td>
<td>NO]</td>
<td></td>
</tr>
<tr>
<td>[.,MAXREGION=nnnn]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[.,MINBUFNO=nnnn]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[.,REREAD]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[.,RLS]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[.,RLSID=cccc]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[.,SHAREOPTION = 1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>[.,TURBO= YES</td>
<td>NO]</td>
<td></td>
</tr>
<tr>
<td>[.,VAROVERFLOW= YES</td>
<td>NO]</td>
<td></td>
</tr>
<tr>
<td>[.,NOMEMLIMIT]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 30-2: Access Override Statement*
DDNAME=

Specifies the DD name of the IAM data set that the override is to be applied to. Up to 40 DD names may be specified. If multiple DD names are specified they must be enclosed in parenthesis and separated by commas.

DD=&ALLDD will result in this override being applied as a global override to all IAM files being accessed in this step, unless otherwise explicitly overridden.

An * as the last character of the specified DDNAME indicates that only the characters up to but not including the * should be used to match the actual DDNAMEs used in this step. This wildcard allows a single override to be applied to more than one DDNAME, based on its name.

There is no default. Either DDNAME or DSN must be specified.

DSN=

Specifies the data set name of the IAM data set that the override is to be applied to. One data set name per override control statement is permitted.

An * as the last character of the specified DSN indicates that only the characters up to but not including the * should be used to match the actual data set names used in this step. This wildcard allows a single override to be applied to more than one data set name.

There is no default. Either DDNAME or DSN must be specified.

ALTLOGGER

When specified, indicates that the IAM/PLEX or IAM/RLS address space is requested to produce replication logger records for the indicated dataset(s). The records will be written to the System Logger logstream identified by the ALTSYSLOG DD card. This override is only honored when specified as an override to an IAM/PLEX or IAM/RLS address space.

AUTOSSYNCPPOINT=

For files that are processed by IAM/RLS or IAM/PLEX, requests that a sync point be taken after the specified number of record lock requesting I/O requests have been issued. Valid values are 0 to 10000. A value of 0 indicates no automatic sync pointing will be done based on I/O for this file. To avoid excessive sync point overhead a nonzero should only be specified for a single file per job step. This override is specified within the JCL for the job step(s) for which this function is desired.

Default is that IAM/RLS or IAM/PLEX sync points will only be taken for batch jobs if explicitly requested.

BACKUPCOMPRESSED

For files that are software compressed, specifies that IAM is not to decompress the data when passing it back to the requester. The data will be passed back to the requester in an IAM data compressed format. The data can be used for backup and reorganization purposes, but it is not usable by application programs. The data must be reloaded into an IAM file with the BACKUPCOMPRESSED IAM CREATE override specified, or the data must be decompressed with the IAMRECVR utility.

Default is that the data is returned in normal, uncompressed format.

BUFSPACE=

Specifies the maximum amount of storage that can be used for buffers, in kilobytes. If specified, this value will supersede any specification for MAXBUFNO. When specified, MAXBUFNO is computed as (BUFSPACE*1024)/BLOCKSIZE.

Default is BUFSP or CICSBUFSP from the IAM Global Options Table.
**BUF64=**

Specifies whether or not 64-bit virtual storage is to be used for the buffers. Valid values are **YES** or **NO**.

Default is **NO** unless the IAM Global Option for BUF64 has been changed.

**CACHE64**

Indicates that IAM should utilize 64-bit addressable virtual storage as a cache area for extended overflow blocks. This may provide for faster sequential processing. Requires z/OS 1.4 and above. Users may have to specify the MEMLIMIT= JCL keyword on the EXEC card to provide 64-bit addressable storage.

**CONSISTENT**

Specified for selected files being used under IAM/RLS or IAM/PLEX for which consistent reads are required. Consistent reads will acquire the record lock to insure that the caller retrieves the most current version of a record. This override is specified on IAM overrides for the IAMRLS address space.

**DEFERWRITE=**

Specifies whether or not IAM will immediately write out updated buffers. Valid values are **YES** or **NO**. Specifying this will override the default for deferred writing, which is based on the share options the file was defined with, and whether or not the file is opened under CICS. Specifying **YES** will indicate that IAM will delay writing out an updated block until the buffer it resides in is required for another block, or until file close. Specifying **NO** will indicate that IAM will always immediately write out any updated data blocks.

Default is **YES** when a file has Share Option of 1 or 2, and is being accessed by a batch job. The default is **NO** when a file has Share Option of 3 or 4. For CICS, all randomly updated, inserted, or deleted records will cause an immediate write unless **YES** is specified. May also default to **NO** based upon the setting for the IMMEDWRITE IAM Global Option.

**DJB=**

Specifies whether or not IAM/RLS and IAM/PLEX will perform Dynamic Job Backout processing for this job. Valid values are **YES** or **NO**. Specifying **YES** will indicate that IAM will backout any records changed by this job step for recoverable IAM files that are processed by IAM RLS should this job step abend. Specifying **NO** will indicate that the backout will not automatically occur; the user will have to use IAMJREST or other recovery process. Only files that have journaling of before images through IAM/RLS and IAM/PLEX will be recovered.

Default is the IAM/RLS and IAM/PLEX Dynamic Job Backout action specified on the IAM RLS startup parameters.
**DYNCORE=**

Specifies an amount of memory for IAM's Dynamic Table. The value is specified in 1K (1024 byte) increments. IAM will attempt to GETMAIN the requested quantity during open, and if successful, will use that storage as a cache for randomly read records. On all random requests, where the complete key is specified, IAM will search the table for the record. These requests are identified on the IAMINFO report by GET RANDOM requests. If found, the record is returned without any physical I/O. If the record is not currently in the table, it will be read from disk, and placed in the table after it is retrieved.

Updates are always made both to the table and to the file. Variable length records are maintained as maximum length entries.

Valid values are from 0 to 16000 (i.e. Dynamic Table up to 16,000K bytes). The storage is requested from extended private. This override cannot be used when DYNDS has been specified.

Default value is 0, the Dynamic Table is disabled.

**DYNDS=**

Specifies the size of the IAM Dynamic Data Space in megabytes (MB). This override cannot be used when DYNCORE is specified. It provides a similar function to the DYNCORE feature, only an MVS Data Space is used instead of virtual storage from the users address space. This will allow for a larger area of storage to be used, with improvements in memory management over DYNCORE.

Valid values are from 32 to 2048.

Default value is 0.

**ESDSLOCK=**

For ESDS type of IAM files, specifies whether or not IAM will use CI locking for applications that are adding records to ESDS files. Valid values are YES or NO. Some applications that write multiple records to an ESDS type of file per transaction may have a requirement that all of those records are written consecutively without being intermixed with records being added by other transactions. That is insured by using the ESDS locking (ESDSLOCK=YES). If the application does not have that dependency, then turning off (ESDSLOCK=NO) this function may yield better performance.

Default is YES unless revised in IAM Global Options Table.

**HPF=**

Specifies whether or not IAM will use z/HPF architecture on devices that are capable of utilizing that function. Specifying YES will cause IAM to use z/HPF architecture on the volume(s) that the system indicates are capable of supporting EXCP processing with z/HPF commands. Specifying NO indicates that IAM will not use the z/HPF architecture I/O commands.

Default is YES that z/HPF will be used unless otherwise specified in the IAM Global Options Table.
INDEXSPACE= Specifies the virtual storage location for the index for the IAM file. Valid values are 64BIT, YES, or NO. If 64BIT is specified the index can reside in 64-bit virtual storage. To use 64-bit virtual storage, the user may need to code the MEMLIMIT= keyword on the EXEC card if their installation does not provide a default amount of 64-bit virtual storage for use. If YES is specified a z/OS Data Space will be used. The size of the Data Space is taken from the IAM Global Options Table DATASPACE value. Only one data space will be obtained per job step; however multiple IAM files are able to utilize the data space. The data space used can grow to the smaller of either 2048 megabytes (2 gigabytes) or 4 times the DATASPACE Global Option value. A specification of NO will result in 31-bit addressable storage within the extended private area of the address space. Use of this capability with either the 64-bit virtual storage or the data space will, for most IAM files, substantially reduce the amount of extended private region used by IAM.

Default is 64BIT if available, otherwise YES unless the installation has changed the IAM Global Option setting.
**JRNAD=**

Specifies the IAM journaling capabilities to be used. If the value specified enables journaling, the value will be combined with any specification from the CREATE override when the file was defined or loaded.

For files not being processed by IAM/RLS or IAM/PLEX users must pre-allocate and catalog a log data set to be used for the journaling, which is required to be the name of the IAM data set appended with the characters "LOG". IAM journaling will not be active during file loads, reorganizations, or during recovery from the journal.

For files being processed by IAM/RLS and IAM/PLEX, the override MUST be provided on the IAM/RLS or IAM/PLEX address space and the users must specify the journal data sets to the IAM/RLS and IAM/PLEX startup procedure if they want to use journaling.

The ACCESS override only changes the journaling value for the job step on which it is specified. Valid values are:

- **BOTH** - The IAM log data set will contain both before and after images. This will enable the user to perform either a forward recovery, or a backward (backout) type of recovery.

- **ALL** - This is a synonym for BOTH. The IAM log data set will contain both before and after images. This will enable the user to perform either a forward or backward (backout) type of recovery.

- **BEFORE** - The IAM log data set will contain before images of updated records. This option allows backward (backout) recoveries only. If AFTER images were specified on a CREATE override, then AFTER images will still be logged to the journal.

- **AFTER** - The IAM log data set will contain after images of updated records. This option allows forward recoveries only. If BEFORE images were specified on a CREATE override they will still be logged to the journal.

- **NONE** - The IAM journaling feature will not be used for the specified IAM data set(s) for this job step.

Default value is NONE, unless otherwise specified.

**LOG=**

Specifies whether or not IAM will display this override card on the job log and in the job messages for this job. Valid values are YES or NO. This is not recommended for use with the REREAD option, particularly on CICS regions, IAM/RLS or IAM/PLEX, as it can result in significant output to the job log and system log.

Default is that the override card is not displayed in the job messages, unless there is a syntax error on the override card.
**MAXBUFNO=** Specifies the maximum number of buffers IAM will acquire during file processing. IAM's Real Time Tuning dynamically adjusts the number of buffers used for a file as I/O demands change, up to the limit specified or defaults to for MAXBUFNO. Valid values are from 1 to 24576.

Default is based on the larger of MAXBUFNO or BUFSP, or for CICS the larger of MAXBUFNO or CICISBUFSP from the IAM Global Options Table. As shipped, the default is the number of buffers that will fit within 65,536K of virtual storage, except for CICS where the default is 1,024K. For a typical IAM file on a 3390, with quarter track blocking, the default MAXBUFNO is 4,904 except under CICS where it is 76.

**MAXREGION=** Specifies the maximum value, in megabytes, to which IAM Dynamic Region Adjustment will set the extended private region. This feature permits processing of files with large virtual storage requirements without the need to modify an installation's IEFUSI exit. Valid values are from 0 to 1024. A value of 0 will disable the Dynamic Region Adjustment feature.

Default value is 512 megabytes, or as otherwise specified in the IAM Global Options Table.

**MAXSECONDARY=** Specifies a multiplication factor to be used by the IAM Dynamic Secondary Space Adjustment feature. Once a file has five or more extents on a volume, the secondary space value will be increased by the MAXSECONDARY factor, providing that it does not exceed the primary space value, or the amount of space available on the volume. Valid values are from 0 to 10. A value of 0 will disable this feature. This is ignored for files in DFSMS Extended Format.

Default value is 5 during file access, or as otherwise specified in the IAM Global Options Table.

**MINBUFNO=** Specifies the minimum number of buffers that IAM will maintain during file processing. IAM's Real Time Tuning dynamically adjusts the number of buffers used for a file as I/O demands change, but will never use less than the specified or defaulted value for MINBUFNO. Valid values are from 1 to 24576. Specifying an equivalent number of buffers for MAXBUFNO will effectively disable IAM's Real Time Tuning algorithm. If MINBUFNO is overridden, then that value will be used as the initial number of buffers to acquire during Open processing. The MINBUFNO will be reduced to MAXBUFNO if the specified MINBUFNO value is larger than MAXBUFNO.

Default value is 1.

**NOMEMLIMIT** Keyword that specifies IAM can ignore the MEMLIMIT for 64-bit virtual storage for the index if necessary to prevent a failure. IAM will only do this when a request is failed due to MEMLIMIT being exceeded up to a limit of 64 gigabytes being used for the index.

Default is IAM abides by the MEMLIMIT, unless otherwise specified in the IAM Global Options Table.

**NORLS** Specifies that IAM is not to use the IAM/RLS and IAM/PLEX services for the identified data set(s).

Default action is based on IAM Global Options, and whether or not the IAM/RLS or IAM/PLEX address space(s) are active.
**REREAD**

Indicates that IAM will reread the overrides each time an IAM file is opened. This is automatically done for CICS, IAM/RLS, and IAM/PLEX address spaces. When used it is recommended that this keyword appear on all of the ACCESS override cards supplied in the file. This is primarily intended to be used by long running jobs where the IAMOVRID points to a sequential data set or PDS member. This provides a mechanism for users to change overrides, and have them take effect by closing and reopening the IAM file(s).

Default is that the IAM override parameters are read when the first IAM file in a job step is opened, and kept in a table in virtual storage.

**REREADEMPTY**

Indicates whether or not IAM is to reread empty prime blocks when processing the file. When IAM is processing a file with REREADEMPTY=NO being specified or defaulted to, IAM will track the blocks that are detected as containing no data. Subsequent requests for those blocks will assume that they are empty, and therefore not do any I/O to read them. With REREADEMPTY=YES, IAM does not keep track of empty blocks, and will always read into storage the requested block.

Default action is based on the Cross Region Share Option, and the type of OPEN. For files defined with Share Option 1 opened for INPUT or UPDATE, and for files defined with Share Option 2 opened for UPDATE, the default is REREADEMPTY=NO. For all other cases, REREADEMPTY defaults to YES.

**RLS**

Specifies that IAM is to process this file using the IAM/RLS or IAM/PLEX services. If the IAM/RLS or IAM/PLEX services are not available, then the OPEN will be failed.

Default is based on the IAM Global Options and whether or not the IAM/RLS or IAM/PLEX address space(s) are active.

**RLSFPSOL**

Specifies the amount of storage to be used, as a multiplier factor to the block size, for the IAM/RLS and IAM/PLEX sequential fast path function. IAM will fill up the specified amount of storage with as many records as will fit each time the request is made. Valid values are 0 – 8. A value of 0 will turn off the Sequential Fast Path processing.

Default value is 2, meaning that IAM will use a buffer area that is double the block size when the IAM/RLS or IAM/PLEX Fast Path function is used. (Note that the Sequential Fast Path function will not be used if the file is opened to a remote IAM/PLEX address space.)

**RLSID=cccc**

Specifies the four character identifier of the IAM/RLS or IAM/PLEX address space that is to be the host address space for this data set. This keyword is primarily intended to be used with a DD=&ALLDD override card, and then with any other override cards that are specified for that job step. Specification of an RLSID on the ACCESS override does not force a dataset to be processed under IAM/RLS or IAM/PLEX, but rather directs it to an IAM/RLS or IAM/PLEX address space if IAM RLS processing is otherwise indicated for this dataset.

Default value is the value of the IAM Global Option RLSID, which is shipped as RLS1.
SHAREOPTION= Specifies the VSAM Cross Region Share Option to be used for this job step. This provides a mechanism for accessing the file with a different Share Option than it had been defined with. Share Options affects the ENQ that is done to protect the file from concurrent access, and will also impact the buffering for the file. Refer to the IDCAMS in Section 10.20 for a description of the various Share Options. Valid values are 1, 2, 3, or 4.

NOTE: SHARING IAM FILES FOR UPDATE IS STRONGLY DISCOURAGED, WITHOUT THE USE OF IAM/RLS, IAM/PLEX, OR OTHER SOFTWARE THAT FACILITATES SHARING! SUCH SHARING MAY ADVERSELY AFFECT THE DATA INTEGRITY OF THE FILE WITHOUT THE PROPER SOFTWARE PROTECTION!

Default is that the file will be opened based on the Share Option specified when the file was defined.

TRACEDDNAME= Activates the IAM request trace capability, and indicates where the trace data is to be written to. Normally, it is written to SYSOUT; however it can also be written to a disk or tape file. To generate a printed report, specify a DD name of IAMTRPRx, where 'x' is any valid character. If 'x' is a 'C', then the key of each record is printed in character format, otherwise the key is printed in hexadecimal format.

WARNING: The trace to a particular output trace data set can only be done from a single IAM file at a time. Please DO NOT SPECIFY TRACEDDNAME WITH DD=&ALLDD. This can result in abends if multiple concurrent traces for different IAM files to the same TRACEDD are being performed.

Default is no trace output is produced.

TRACEREQUEST= Specifies the type of tracing to be done.

TRALL - Activates all of the trace points.
TRIOS - Trace start of each logical I/O request.
TRIOE - Trace at end of each logical I/O request.
TRBFR - Trace internal IAM calls to the IAM Buffer Manager.
TREXCP - Trace physical I/O requests.
TRXTND - Trace file expansion (extend) calls.
TRDYN – Trace calls to the Dynamic Data Space.
TRINT -- Tracing is done to an internal table kept in 64bit virtual storage.

Multiple values for TRACEREQUEST can be specified by enclosing the desired keywords within parenthesis.

Default value is TRIOS.
**TURBO=**

Specifies whether or not IAM should modify its Real Time Tuning algorithms when heavy physical I/O is being performed in short time periods. Valid values are **YES** or **NO**.

The default value is **YES**, unless changed in the IAM Global Options Table. This global option is set via the **ENABLE** and **DISABLE** Global Options.

**UPDATENQ=**

Overrides the ENQ processing that IAM will perform for the specified file. Normally, IAM will base the ENQ on the defined cross region share option. Valid values are:

- **EXCL** - Indicates that IAM is to perform an exclusive ENQ on the file. This will prevent any other OPEN for this IAM file, within the scope of the ENQ capabilities provided on the host system.

- **SHR** - Indicates that IAM is to perform a shared ENQ on the file. If specified for all users of the file, it will permit multiple jobs to have update capability to the same IAM file.

- **NONE** - Prevents IAM from issuing any ENQ for the file being opened. The file will not be protected from concurrent update or even from a concurrent attempt to reload the file, while this application has the file open. This value should be specified when the file is being used by a job that is utilizing MVS Checkpoint / Restart facilities.

The default ENQ processing is based on the cross region share options, and the type of OPEN being issued for the data set.

**VAROVERFLOW=**

Specifies whether or not IAM is to enable variable length record support for the overflow area. This override will only take effect when the file is opened for update. **Once a file has variable overflow enabled, it cannot be turned off until the file is reorganized.** Valid values are **YES** or **NO**.

Default is the value set when the file is defined or loaded, which defaults to **YES**, unless changed in the IAM Global Options Table.
30.04 IAM Override Statement Examples

The examples in this section are intended to help demonstrate how to use the IAM Override capability. There are other examples of using IAM Overrides throughout the manual, as the various features and capabilities of IAM are presented.

Example 1: Define with Override

The following example shows how an IAM Override statement is used during an IDCAMS DEFINE of an IAM file, to request the file's data records be compressed. The IAM file's DEFINE will use the attributes of an existing VSAM cluster.

```
//DEFINE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//IAMOVRID DD *
CREATE DD=ALLDD,DATACOMPRESS=YES
/*
//SYSIN DD *
DEFINECLUSTER -(NAME(CICS.MASTER.$IAM) -
MODEL(CICS.MASTER.CLUSTER) -
VOLUMES(MVS001) )
/*
```

Figure 30-3: Example of IAM Override on Define
Example 2: Multiple Overrides on Define

The following example shows an IDCAMS DEFINE of multiple IAM files, each with a different IAM Override values. As seen, this can easily be done through the use of the DSN parameter on the IAM Create Override statement.

The data set name “iam.master” will be a data compressed file, with Automatic Space Release turned off, due to anticipated file expansion. The data set's block size will be based on the IAM default value.

The data set named “iam.test” will use 1/2 track blocking, as indicated by the B=2 override. It will use the default for data compression and space release, as indicated in the IAM Options Table.

```
//DEFINE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=* 
//IAMOVRID DD *
   CREATE DSN=iam.master,DATACOMPRESS=YES,RELEASE=NO
   CREATE DSN=iam.test,B=2
/*
//SYSGIN DD *
DEFINE CLUSTER -
   (NAME(iam.master) -
   OWNER($IAM) -
   CYLINDERS(100 20) -
   FREESPACE(10 5) -
   KEYS(24 8) -
   RECORDSIZE(250 480) -
   VOLUMES(IAM001) )
DEFINE CLUSTER -
   (NAME(iam.test) -
   OWNER($IAM) -
   CYLINDERS(10 5) -
   FREESPACE(10 5) -
   KEYS(8 8) -
   RECORDSIZE(100 1250) -
   VOLUMES(IAM001) )
/*
```

Figure 30-4: Example of Multiple Overrides on IDCAMS Define of Multiple Data Sets
Example 3: File Load Override

The following example demonstrates the use of an IAM Override for a file load, which is being done through IDCAMS. In particular, the CRBUFOPT override is specified to provide maximum buffering on the IAM file.

```
//LOADIAM EXEC PGm=IDCAMS
//SYSPRINT DD SYSOUT=* 
//IAMINFO DD SYSOUT=* 
//IAMOVRID DD *
CREATEDD=IAMFILE,CRBUFOPT=MCYL
/* 
/IAMFILE DD DISP=OLD,DSN=my.iam.file
/INPUTFIL DD DSN=my.sequential.file,DISP=OLD,
// DCB=BUFNO=30
//SYSPIN DD *
REPRO INFILE(INPUTFIL) OUTFILE (IAMFILE) REUSE
/*
```

Figure 30-5: Example of Overriding a File Load

Using IAM Overrides with CICS

The following example shows IAM Overrides for files being processed under CICS. The IAM override cards are contained within a partitioned data set (PDS). The PDS must be defined with fixed length records of 80 bytes. The data set can be blocked at any valid block size for that record size. IAM will request that the buffers that are acquired by MVS for reading the override data set are to reside in extended private. The IAMOVRID and IAMINFO files do NOT have to be defined to CICS.

In order to provide increased flexibility in IAM Version 9.2 the override input is always reread for each dataset that is opened. This allows IAM to pick up any changes to the overrides without having to recycle CICS.

The following override values are being specified. The MAXBUFNO value for files other than those otherwise overridden is being set to 16. Two of the files will have their maximum buffers increased to 128 buffers. Another file will use the Dynamic Table (DYNCORE) option to table records in memory. A size of 1024K of memory is specified for the table. Also note that an IAM CREATE override is provided specifying CRBUFOPT=TRK. This is recommended to minimize the amount of storage used just in case an IAM file is opened empty under CICS, because the file load may use significant virtual storage resources. The contents of the control card file, for the above mentioned overrides are as follows:
Example 4: CICS Overrides in a PDS

<table>
<thead>
<tr>
<th>IAM Override Statement Examples 30.04</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAM Override Statement Examples 30.04</td>
</tr>
</tbody>
</table>

Below is an example of the JCL that could be used by CICS to utilize the above IAM Overrides. CICS can be executed here as either a started task, or as a batch job. Note that for started tasks, the IAM Overrides cannot reside in an in-stream data set, i.e. (DD *) is not allowed. Also note that the IAMOVRID is specified with a DISP=SHR, which will allow the file to be updated while CICS is running. While the IAM files here are shown as being allocated at CICS startup, the files could also be dynamically allocated by CICS.

```
DSN=my.iam.override(cics1):

  ACCESS DD=(FILE1,FILE2),MAXBUFNO=128
  ACCESS DD=FILE3,DYNCORE=1024
  CREATE DD=&ALLDD,CRBUFOPT=TRK
  ACCESS DD=&ALLDD,MAXBUFNO=16
```

Figure 6: Example of IAM Override cards in a partitioned data set

Example 5: CICS JCL with Overrides

```
//CICS1 EXEC PGM=DFHSIP
//.....
//FILE1 DD DSN=IAM.FILE1,DISP=SHR
//FILE2 DD DSN=IAM.FILE2,DISP=SHR
//FILE3 DD DSN=IAM.FILE3,DISP=SHR
//.....
//IAMINFO DD SYSOUT=*  
//IAMOVRID DD DISP=SHR,DSN=my.iam.override(cics1)
```

Figure 7: Example of a CICS PROC/JCL with IAM Overrides in a PDS
When using IAM data sets with alternate indexes and paths, the DSN= parameter must be used, instead of the DD= parameter. This is because the actual IAM data sets that are used will be dynamically allocated with system generated DD names, so the DD names are not known in advance. The only meaningful override for a PATH is the trace information. If tracing is specified for a PATH, tracing will be propagated to all of the data sets opened because of the PATH open. Other overrides on the PATH will be ignored.

The example below shows how to specify an override for a base cluster and an alternate index, which are being accessed via a PATH. The override for the base cluster will specify a MAXBUFNO value, while the override for the alternate index is using the IAM Dynamic Tabling feature. Use of dynamic tabling on the alternate index will provide a way to keep the alternate index in virtual storage, if it is not too large, and if it is being randomly accessed.

```
//UPDATEAX EXEC PGM=UPDATE
//MASTRPTH DD   DSN=iam.master.path,DISP=OLD
//IAMOVRID DD   *
ACCESS DSN=iam.master.cluster,MAXBUFNO=128
ACCESS DSN=iam.master.aix,DYNCORE=1024
/*
```

*Figure 30-6: Example of Overrides with an IAM Alternate Index*
IAMSMFVS is a special purpose SMF data analysis program, intended to provide useful information in a concise format about VSAM and IAM data set activity. The information can be used to:

1. Identify VSAM Clusters that are candidates for conversion to IAM.
2. Compare the results once the files are converted to IAM.
3. As a source for tracking IAM data set activity to monitor IAM data set usage.
4. Determine when IAM files need to be reorganized.

IAMSMFVS provides four different reports and an optional CSV (comma separated value) file. The first report is a Summary report, which includes a summarization of VSAM and IAM activity so it is easy to tell the DASD space and I/O totals for each of the indexed access methods. The second report is an EXCP report, which is broken down by access method. There is a VSAM EXCP Report, and an IAM EXCP Report. Each report has the top 100 data sets for I/O activity for that access method. There is an option, DETAIL, which will provide a line for each SMF record for each of the data sets included in the EXCP report, with a breakdown of activity for that job step. The third report is a Data Set Summary Report, which contains information about each data set for which information was gathered by IAMSMFVS sorted by the data set name. The last report is based on data set size, here again split by access method. Each size report has the largest 100 data sets for the specified access method.

The optional CSV file contains information on the data sets that are in the VSAM EXCP REPORT. It includes information that is taken from both the EXCP report and the Data Set Summary report. This provides a convenient way to work with and review the data for those data sets which are likely to provide the most performance improvement by conversion to IAM. The CSV file should be included with the reports when sent to Innovation for analysis.

The IAMSMFVS program analyzes data set usage statistics gathered primarily from the Systems Management Facility (SMF records), with supporting data from the system catalog. SMF data may be taken from either the active SMF data sets or sequential SMF history data sets. To operate, IAMSMFVS requires SMF type 30 records with step end and interval type records. For reporting on VSAM data sets the SMF type 64 records must be collected. For reporting on IAM data sets, the IAM SMF recording must be enabled and an SMF record type selected in the IAM Global Options table. IAMSMFVS requires matching Step Termination or Interval record for each data set record; otherwise the data set record will not be processed. If the required SMF record types are not present, or have been modified from the standard z/OS format, then IAMSMFVS will not be able to produce any reports.

IAMSMFVS should be run, if at all possible, on an LPAR or System that has access to the catalog(s) for the VSAM data sets that are contained in the SMF data being processed. This enables IAMSMFVS to report on information that is not in the SMF data, including cluster attributes such as type of cluster (i.e. KSDS, ESDS, RRDS), defined average record length, key offset, and free space specifications. While IAMSMFVS will use the catalog information to relate the different VSAM components, although it does attempt to do so from the data in the SMF records if the cluster is not found in the catalog. IAMSMFVS use of the catalog can be turned off with the NOLOCATE keyword on the REPORT control card.
IAMSMFVS JCL Requirements

The JCL statements required to execute IAMSMFVS are as follows:

EXEC Statement
Specifies the IAM VSAM SMF ANALYSIS program name - IAMSMFVS. A large region parameter may need to be provided, based on the number of data sets that are being tracked. IAMSMFVS uses above and below the 16-megabyte line storage.

DD Statements
The following table describes the required DD statements for running IAMSMFVS. The amount of DASD space required for the temporary work files of SORTIN, SORTOUT, and SORTWKxx will depend entirely on the amount of SMF data that is being processed. A month’s history tape is going to require much more DASD space than just a daily snapshot.

<table>
<thead>
<tr>
<th>DD Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB or JOBLIB</td>
<td>Specifies the IAM Load Module Library. This statement may be omitted if the IAM modules are in a link list, as is recommended.</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>Specifies where the IAMSMFVS reports are to be printed. Usually a SYSPRINT data set.</td>
</tr>
<tr>
<td>SYSMF</td>
<td>Identifies the file containing the input SMF data. This may be an active SMF dataset or any file containing off loaded SMF data.</td>
</tr>
<tr>
<td>SYSIN</td>
<td>Specifies the control card input data set. Usually a DD * data set.</td>
</tr>
<tr>
<td>SORTIN</td>
<td>Specifies a work file that contains extracted SMF data that is used by IAMSMFVS to create the reports. This data will be passed to the external sort.</td>
</tr>
<tr>
<td>SORTOUT</td>
<td>Specifies a work file, which will be returned from the external sort routine.</td>
</tr>
<tr>
<td>SORTWKnn</td>
<td>If there are any DD statements needed by your sort program (i.e. SORTLIB, SORTWKnn, etc.), they must be included. Refer to documentation for your sort.</td>
</tr>
<tr>
<td>SORTLIB</td>
<td>SortLIB</td>
</tr>
<tr>
<td>VSAMCSV</td>
<td>An optional DD card that when present will contain information on the VSAM data sets in comma separated format (CSV) for easy use in a spread sheet. There will be a row for each VSAM cluster that are in the VSAM EXCP REPORT, and will contain information that is in the EXCP report and Data Set Summary reports. It is to be a sequential data set with no DCB characteristics specified. Only a few tracks are necessary to be allocated for this data set.</td>
</tr>
</tbody>
</table>
The REPORT command is used to initiate IAMSMFVS processing. There are various options that can be specified, as described below. The input can be an SMF history file (RECFM=VBS) or a system SMF data set. It is recommended that initially no operands be specified. The default reports would then list the 100 data sets, by data set organization, with the most EXCP activity, and the 100 largest data sets, again by data set organization.

**Operands**

The following operands may be specified on the REPORT command. The underscored portion indicates the minimum abbreviation for the keyword.

<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATONLY</td>
<td>Specifies that IAMSMFVS will only report on VSAM data sets that are cataloged. This will eliminate from the report stray component entries for VSAM data sets that are not currently cataloged.</td>
</tr>
<tr>
<td>CHECKLENGTH</td>
<td>Specifies that SMF data records are to be validated against a table of minimum record lengths. If the user has modified the minimum record length or is executing in a non-compatible system, the correct record lengths may be specified by the RECSIZE operand. By default, the length of an SMF record is not validated.</td>
</tr>
<tr>
<td>CURRENT</td>
<td>Specifies that only the most recent values for overflow use in IAM files will be reported on, rather than the maximum amount used. This may be useful for determining when an IAM file needs to be reorganized. By default, the maximum amount of overflow use is reported on.</td>
</tr>
<tr>
<td>DETAIL</td>
<td>The EXCP Reports are to include detailed statistics by job step. By default, only summary data set activity is reported.</td>
</tr>
</tbody>
</table>

Figure 40-1: Report Command Operands
**Operand** | **Description**
--- | ---
**DFEFERRPRT=** | Indicates whether or not error messages from the VSAM ICF identification processor should be printed. Valid values are:

- **AC** - All messages to the console
- **NO** - Error messages are not to be printed.
- **YES** - Error messages are to be printed.

The default is **NO**.

**DSGROUPS=** | Specifies that only records having data set names which begin with the given character string(s) will be processed. This operand specifies a partial dataset name from 1 to 44 characters in length. Up to 100 data set groups may be specified for a single command if entered as follows:

**DSGROUPS=(dsg1,dsg2,...,dsgn)**

If neither the DSGROUPS nor DSNAMES operand is specified, data set selection will be based upon the value specified for DSORG (or its default).

**DSNAMES=** | Specifies that only records having a data set name which match the dataset name(s) specified will be processed. This operand specifies a complete data set name from 1 to 44 characters in length. Up to 100 dataset names may be specified for a single command if entered as follows:

**DSNAMES=(c...c,...,c...c)**

If neither the DSGROUPS nor DSNAMES operand is specified, data set selection will be based upon the value specified for DSORG (or its default).

**DSORG=** | Identifies the data set organization that is to be processed. Valid values are:

- **AM** - only VSAM clusters.
- **IAM** - only IAM data sets.

The default is **AM (VSAM) and IAM**.

**NOTE:** To extract data for IAM data sets, SMF recording of IAM SMF records must be enabled, as described in the IAM Users Manual in Section 91.

**ERRORPRINT** | Specifies that any SMF data record that causes an error during processing or fails length verification is to be printed.

The default is records in error are not printed.

**FROMDATE=** | Specifies the lower date limit of the SMF records that are to be analyzed, in the form 'yyyyddd'.

The default is that there is no lower limit on the date for record selection.

**FROMDDNAME=** | Specifies the DDNAME of the SMF file that is to be used as input to IAMSMF.

The default input DDNAME is **SYSMF**.
**Operand**

**FROMTIME=**

Specifies the lower time limit of the SMF records that are to be analyzed, in the form of 'hh:mm:ss'.

When used with FROMDATE, forms a combined starting point of date and time.

When used without FROMDATE, the FROMTIME applies to all days for which SMF records are being processed.

The default is there is no lower time limit on the record selection.

**GROUPNAMES=**

Specifies that only those SMF records having a job name which begin with the specified character string(s) will be analyzed. This operand specifies a partial job name from 1 to 8 characters in length. Up to 50 job groups and/or names may be specified for a single command if entered as follows:

GROUPNAMES=(jobn1,jobn2,jobn3,.....,jobnx)

The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the job name will not participate in SMF record selection.

**JOBNAMES=**

Specifies those only records having a jobname which match the jobname(s) specified will be copied. This operand specifies a complete jobname from 1 to 8 characters in length. Up to 50 names may be specified in a single command if entered as follows:

JOBNAMES=(jobname1,jobname2,.....,jobnamex)

The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the job name will not participate in SMF record selection.

**MAXDSENS=**

Specifies the maximum number of unique data sets and/or clusters that will be tabled during this execution. The number can be any value from 20 to 5,000,000 inclusive.

The default is 60,000 data set table entries.

**MAXJOBS=**

Specifies the maximum number of unique job names that will be tabled during this execution. The number can be any value from 20 to 5,000,000 inclusive.

The default is 50,000 job name table entries.

**MAXRECLength=**

Specifies the largest SMF record that the program will process. The number may be any value from 16384 to 65536, inclusive.

The default is 32768 bytes.

**MAXREPORTS=**

Specifies the maximum number of data set and/or cluster names to be included in the EXCP Activity Report(s) and the Size Report(s). The number can be any value from 20 to 32000, inclusive. All data sets are included in the data set report.

The default is 100.

**NOLOCATE**

Indicates that no catalog locates will be issued for VSAM data sets. This will leave some fields blank in the reports, but avoids potential catalog errors.
**Operand** | **Description**
--- | ---
**PRTLENGTH=** | Limits the number of bytes of data to be printed if ERRORPRINT is indicated. The number may be any value from 32 to 65536, inclusive.

The default is 32768 bytes.

**RECSIZE(rrr)=** | Establishes the minimum length of the SMF record type 'rrr' as the value 'nnnn'.

The default minimum record lengths for system generated SMF records are documented in the IBM Systems Management Facilities manual.

**SORTCORE=** | Specifies the amount of storage for the SORT to use. The number maybe any value from 10000 to 8000000 inclusive.

The default is 100000.

**SORTMSG=** | Specifies the message option to be used by the SORT. Valid values are:

- **AC** - All messages to the console
- **AP** - All messages to the printer (SYSOUT)
- **CC** - Critical messages to the console
- **CP** - Critical messages to the printer
- **NO** - No messages to be produced
- **PC** - Critical messages to both console and printer

The default is CC.

**SORTPFX=** | Specifies the DDNAME prefix to be used by the SORT. If the string specified is less than 4 characters, a dollar sign ($) fill character will be used.

The default is SORT.

**TEMPDSNAMES** | Specifies that reports produced by the REPORT command will contain information on temporary as well as permanent data sets.

By default, temporary data sets are ignored.

**TODATE=** | Specifies the upper date limit of the SMF records that are to be analyzed. Must be in the form 'yyyyddd'.

The default is that there is no upper date limitation for record selection.

**TOTIME=** | Specifies the upper time limit of the SMF records that are to be analyzed, in the format of 'hh:mm:ss'.

When used with the TODATE keyword, this forms a combined ending point of the date and time specified. When used without the TODATE keyword, the 'TOTIME' is applied to each day for which SMF records are being processed.

The default is that there is no maximum time.
40.04 IAMSMFVS Usage Examples

Shown below are some examples of how to run IAMSMFVS, which include JCL and the control card input. Examples and descriptions of the reports produced are in the following sections. We recommend producing reports on at least seven days of data.

**Example A:** Basic IAMSMFVS Report

The first example demonstrates how to run IAMSMFVS to obtain the basic reports, which will include the 100 busiest VSAM data sets, and the 100 largest VSAM data sets. A CSV file will be produced to the dataset specified by the VSAMCSV DD card, which can be loaded into a spreadsheet for easier review and analysis. IAM data set reports will also be produced, providing that the IAM SMF recording has been activated.

```
//REPORT   EXEC PGM=IAMSMFVS,REGION=0M
//STEPLIB  DD   DISP=SHR,DSN=iam load library
//SYSPRINT DD   SYSOUT=*  
//SYSMF    DD   DISP=SHR,DSN=smf data
//SORTIN   DD   UNIT=SYSDA,SPACE=(CYL,(100,20))
//SORTOUT  DD   UNIT=SYSDA,SPACE=(CYL,(100,20))
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(100,20))
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(100,20))
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(100,20))
//VSAMCSV  DD   DSN=my.vsamcsv.csv,UNIT=SYSDA,
// DISP=(,CATLG),SPACE=(TRK,(2,1))
//SYSIN    DD   *
//REPORT MAXDSNS=99999,MAXREPORTS=300
/*
```

*Figure 40-2: Example 1 of Using IAMSMFVS (EX4004A)*

**Example B:** Reporting on More Data Sets

This next example demonstrates obtaining the IAMSMFVS reports with providing a larger capacity both in the number of data sets tabled (MAXDSNS=90000), and in the number of data sets included in the EXCP and Size Reports, (MAXREPORTS=300).

```
//REPORT   EXEC PGM=IAMSMFVS,REGION=0M
//STEPLIB  DD   DISP=SHR,DSN=iam load library
//SYSPRINT DD   SYSOUT=*  
//SYSMF    DD   DISP=SHR,DSN=smf data
//SORTIN   DD   UNIT=SYSDA,SPACE=(CYL,(100,20))
//SORTOUT  DD   UNIT=SYSDA,SPACE=(CYL,(100,20))
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(100,20))
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(100,20))
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(100,20))
//VSAMCSV  DD   DSN=my.vsamcsv.csv,UNIT=SYSDA,
// DISP=(,CATLG),SPACE=(TRK,(2,1))
//SYSIN    DD   *
//REPORT MAXDSNS=90000,MAXREPORTS=300
/*
```

*Figure 40-3: Example of Reporting on More VSAM Data Sets (EX4004B)*
Example C: Using a Date Range

In the next example, an SMF history tape is being processed. The FROMDATE and TODATE keywords are specified, to limit the report to only one particular week's worth of data.

```plaintext
//REPORT EXEC PGM=IAMSMFVS,REGION=4M
//STEPLIB DD DISP=SHR,DSN=iam load library
//SYSPRINT DD SYSOUT=* 
//SYSMF DD DISP=SHR,DSN=smf history data
//SORTIN DD UNIT=SYSDA,SPACE=(CYL,(100,20))
//SORTOUT DD UNIT=SYSDA,SPACE=(CYL,(100,20))
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(100,20))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(100,20))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(100,20))
//VSAMCSV DD DSN=my.vsamcsv.csv,UNIT=SYSDA,
// DISP=(,CATLG),SPACE=(TRK,(2,1))
//SYSIN DD *
REPORT FROMDATE=2014230,TODATE=2014237
/*
```

Figure 40-4: IAMS MFVS Example with FROMDATE and TODATE (EX4004C)

Example D: Reporting on IAM Data Sets

The following example demonstrates how to generate a report to help determine which IAM data sets need to be reorganized. The IAM SMF recording must have been enabled so that there are SMF records to report on. To make sure that the most recent information is being used, the CURRENT keyword is specified. A FROMDATE is also being specified, as the input is an SMF history tape. The IAMS MFVS report that will be most useful in determining if reorganization is necessary is the IAM Size report. MAXREPORTS is being specified, because there are more than 100 IAM data sets in the shop.

```plaintext
//REPORT EXEC PGM=IAMSMFVS,REGION=4M
//STEPLIB DD DISP=SHR,DSN=iam load library 
//SYSPRINT DD SYSOUT=* 
//SYSMF DD DISP=SHR,DSN=smf history data
//SORTIN DD UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTOUT DD UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50,10))
//SYSIN DD *
REPORT FROMDATE=2014230,DSORG=IAM,CURRENT, MAXREPORTS=500
/*
```

Figure 40-5: Reporting on IAM Data Sets (EX4004D)
IAMSMFVS supplies the user with a SUMMARY and three reports. Those reports are the EXCP Report, the Data Set Summary Report, and the Size Report. Examples of each of the reports, along with explanations of the fields are presented below.

**Summary Report**

On the very first page, the REPORT control card being processed is printed. That is followed by summary information for the run.

**IAM601 Message**

The IAM601 message presents a summary of the input SMF data. This includes the number of records read from the input SMF file, the number of records used, and the number of records dropped. Used records indicate the number of Step termination records (type 30 subtype 4 or type 4) and the number of data set records (IAM records, VSAM type 64 records) that met the selection criteria for this execution. The number of records dropped indicates the number of records that exceeded the maximum record length.

The following line indicates the number of data sets (VSAM and/or IAM) that were included in the table for analysis and the number of unique job names that were found. (Note that this is not an actual count of number of jobs, only of the number of unique job names.) This is followed by the date and time period from the SMF records that was found on the input file.

**Space Utilization Summary**

The next set of information provided is a space utilization summary, based on DASD device type and data set organization. This section will only include data for the data set organizations that IAMSMFVS was processing data for. So, for example if you had requested only VSAM data sets (DSORG=AM), then no values would be provided for IAM data sets. The amount of space figure is based upon the maximum space for each data set, as determined from the information in the SMF records.

**Total Disk EXCP’s**

Then an EXCP summary is provided. The Total Disk EXCP's comes from an accumulation of the information from the step termination records for all non-temporary DASD data sets. The figures for each access method are the totals from the actual data set SMF records.
After the summary report appear the EXCP reports. These reports are organized by access method, and contain the data sets with the most EXCP's. The maximum number of data sets in each access method report is based on the MAXREPORTS keyword of the REPORT command, which defaults to 100. The different access method reports are quite similar, so samples of VSAM and IAM EXCP Reports will be shown, followed by descriptions of the various fields contained in the report.

### VSAM EXCP Report

<table>
<thead>
<tr>
<th>DATA SET NAME</th>
<th>%CLUSTER</th>
<th>TOTAL EXCPS</th>
<th>TOTAL RECORDS</th>
<th>READS</th>
<th>INSERTS</th>
<th>UPDATES</th>
<th>DELETES</th>
<th>SPLIT</th>
<th>ALLOC</th>
<th>CI</th>
<th>CA</th>
<th>TRKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAMV.BNC750V.CLUSTER</td>
<td>100</td>
<td>16481568</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC750V.DATA</td>
<td>78</td>
<td>12789824</td>
<td>2000000</td>
<td>214400000</td>
<td>497504</td>
<td>3200000</td>
<td>497504</td>
<td>14016</td>
<td>3184</td>
<td>24</td>
<td>135</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC750V.INDEX</td>
<td>22</td>
<td>3691744</td>
<td>2816</td>
<td>0</td>
<td>0</td>
<td>86296</td>
<td>0</td>
<td>3184</td>
<td>140</td>
<td>318</td>
<td>24</td>
<td>135</td>
</tr>
<tr>
<td>IAMV.BNC1KV.CLUSTER</td>
<td>100</td>
<td>16429455</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC1KV.DATA</td>
<td>79</td>
<td>13032467</td>
<td>2000000</td>
<td>187600000</td>
<td>435316</td>
<td>2800000</td>
<td>435316</td>
<td>11240</td>
<td>3605</td>
<td>30</td>
<td>135</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC1KV.INDEX</td>
<td>21</td>
<td>3396988</td>
<td>3621</td>
<td>0</td>
<td>0</td>
<td>86296</td>
<td>0</td>
<td>3605</td>
<td>112</td>
<td>318</td>
<td>24</td>
<td>135</td>
</tr>
<tr>
<td>IAMV.BNC500V.CLUSTER</td>
<td>100</td>
<td>12599784</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC500V.DATA</td>
<td>73</td>
<td>9237824</td>
<td>2000000</td>
<td>214400000</td>
<td>497504</td>
<td>3200000</td>
<td>497504</td>
<td>46424</td>
<td>2080</td>
<td>24</td>
<td>135</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC500V.INDEX</td>
<td>27</td>
<td>3361980</td>
<td>1813</td>
<td>0</td>
<td>0</td>
<td>85008</td>
<td>0</td>
<td>3605</td>
<td>112</td>
<td>318</td>
<td>24</td>
<td>135</td>
</tr>
<tr>
<td>IAMV.BNC250V.CLUSTER</td>
<td>100</td>
<td>10238592</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC250V.DATA</td>
<td>70</td>
<td>7187432</td>
<td>1813</td>
<td>0</td>
<td>0</td>
<td>50664</td>
<td>0</td>
<td>2080</td>
<td>112</td>
<td>318</td>
<td>24</td>
<td>135</td>
</tr>
<tr>
<td>IAMV.BNC250V.INDEX</td>
<td>30</td>
<td>3051160</td>
<td>877</td>
<td>0</td>
<td>0</td>
<td>35096</td>
<td>0</td>
<td>976</td>
<td>112</td>
<td>318</td>
<td>24</td>
<td>135</td>
</tr>
</tbody>
</table>

**Figure 40-7: Sample VSAM EXCP Report**

### IAM EXCP Report

<table>
<thead>
<tr>
<th>DATA SET NAME</th>
<th>USE COUNT</th>
<th>TOTAL EXCPS</th>
<th>TOTAL RECORDS</th>
<th>READS</th>
<th>INSERTS</th>
<th>UPDATES</th>
<th>DELETES</th>
<th>OVERFLOW RECS</th>
<th>USED TRKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAMV.BNC1KI.CLUSTER</td>
<td>343</td>
<td>4070498</td>
<td>236087152</td>
<td>652974</td>
<td>2100000</td>
<td>652974</td>
<td>20051</td>
<td>41438</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC750I.CLUSTER</td>
<td>391</td>
<td>2922126</td>
<td>260104976</td>
<td>715162</td>
<td>2300000</td>
<td>715162</td>
<td>15808</td>
<td>31009</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC500I.CLUSTER</td>
<td>374</td>
<td>1659165</td>
<td>248796064</td>
<td>684068</td>
<td>2200000</td>
<td>684068</td>
<td>14828</td>
<td>20664</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC250I.CLUSTER</td>
<td>360</td>
<td>777792</td>
<td>261813888</td>
<td>746256</td>
<td>2400000</td>
<td>746256</td>
<td>15055</td>
<td>10335</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC125I.CLUSTER</td>
<td>315</td>
<td>359822</td>
<td>229087152</td>
<td>652974</td>
<td>2100000</td>
<td>652974</td>
<td>13032</td>
<td>5218</td>
<td></td>
</tr>
<tr>
<td>IAMV.TEST.CLUSTER</td>
<td>31</td>
<td>214495</td>
<td>1366072</td>
<td>520000</td>
<td>120000</td>
<td>16412</td>
<td>30</td>
<td>5469</td>
<td></td>
</tr>
<tr>
<td>IAMV.LRG922.CLUSTER</td>
<td>8</td>
<td>1132</td>
<td>1000005</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>1857</td>
<td></td>
</tr>
<tr>
<td>IAMV.TEST0.CLUSTER</td>
<td>3</td>
<td>971</td>
<td>306980</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>4899</td>
<td></td>
</tr>
<tr>
<td>IAMV.LRG923.CLUSTER</td>
<td>4</td>
<td>627</td>
<td>400002</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>1857</td>
<td></td>
</tr>
<tr>
<td>IAMV.JFM600.CLUSTER</td>
<td>6</td>
<td>136</td>
<td>25000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>313</td>
<td></td>
</tr>
<tr>
<td>IAMV.JFM600.NEW.CLUSTER</td>
<td>5</td>
<td>15</td>
<td>25000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>313</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 40-8: Figure 9: Sample IAM EXCP Report**
A description of the fields appearing on the EXCP reports is provided below. Note that some fields represent an accumulated value, whereas other fields indicate a maximum value. For IAM, maximum values for overflow are replaced with the most recent values if the keyword CURRENT is specified on the REPORT command.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Set Name</td>
<td>Specifies the name of the data set. For cataloged VSAM data sets, this will have the cluster name on the first line, followed by the component names on subsequent lines, which are indented.</td>
</tr>
<tr>
<td>% Cluster EXCP’s</td>
<td>Indicates the percentage of the total EXCP’s of each of the components of the VSAM cluster. For the cluster itself this value will be 100. Due to rounding, the total of the components may not add up to 100. (This is in the VSAM EXCP report only.)</td>
</tr>
<tr>
<td>Use Count</td>
<td>Indicates the number of SMF records processed for this data set. This will usually be the number of times the data set was closed during the time interval of the report. (This column is only in the IAM EXCP report.)</td>
</tr>
<tr>
<td>Total EXCPS</td>
<td>The accumulated total EXCP counts from all of the SMF records processed for this data set. For cataloged VSAM clusters, this number will be the total for all of the components, with the component value given on the subsequent lines.</td>
</tr>
<tr>
<td>Records</td>
<td>Contains the maximum number of records that were in the file.</td>
</tr>
<tr>
<td>Reads</td>
<td>Accumulated total from all of the SMF records found for the specified data set, of the records read.</td>
</tr>
<tr>
<td>Inserts</td>
<td>Accumulated total number of records inserted. This value does NOT include the count of records from an initial file load. Also, for VSAM clusters, does not include records added to the end of the file.</td>
</tr>
<tr>
<td>Updates</td>
<td>Accumulated total number of records updated.</td>
</tr>
<tr>
<td>Deletes</td>
<td>Accumulated total number of records deleted.</td>
</tr>
<tr>
<td>Splits - CI</td>
<td>For VSAM components, the accumulated total number of CI splits that occurred.</td>
</tr>
<tr>
<td>Splits - CA</td>
<td>For VSAM components, the accumulated total number of CA splits that occurred.</td>
</tr>
<tr>
<td>Alloc Tracks</td>
<td>For VSAM components, the maximum number of tracks allocated to the component.</td>
</tr>
<tr>
<td>Overflow Recs</td>
<td>For IAM data sets, indicates the maximum number of records in overflow, unless the keyword CURRENT was specified on the REPORT command. In that case, it will contain the value from the most recent SMF record for the data set.</td>
</tr>
<tr>
<td>Overflow %</td>
<td>For IAM data sets, indicates the maximum percentage, or most recent value if CURRENT was specified, of the overflow area that was used. If a file has no overflow area, then this number will be 100 even though there are no records in the overflow area. For Enhanced format files that are defined with an Overflow override value, the percentage is based on that number. For Enhanced format files without an Overflow override, value is based on assumption using a total value for overflow from overflow blocks currently used plus all of the unused extended blocks. For Compatible format files, this is the percentage is based on the established size of the Independent Overflow Area.</td>
</tr>
<tr>
<td>Used Tracks</td>
<td>For IAM data sets, indicates the maximum amount of DASD space used for this file.</td>
</tr>
</tbody>
</table>
**40.07 IAMSMFVS Data Set Summary Report**

After the EXCP reports IAMSMFVS produces a Data Set Summary Report. This report contains all of the data sets that were encountered in the SMF data, and retained in the data set table built by IAMSMFVS. Data sets appear in ascending name sequence. For cataloged VSAM clusters, they are sorted based on the cluster name.

### Data Set Summary Report

<table>
<thead>
<tr>
<th>DATA SET NAME</th>
<th>USE</th>
<th>TOTAL</th>
<th>EXCPS</th>
<th>DSORG</th>
<th>RECFM</th>
<th>LRECL</th>
<th>LEN</th>
<th>RKP</th>
<th>CISIZE</th>
<th>CI%</th>
<th>CA%</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAMV.BNC1KI.CLUSTER</td>
<td>343</td>
<td>4070498</td>
<td>IAM</td>
<td>VO</td>
<td>1000</td>
<td>1000</td>
<td>8</td>
<td>0</td>
<td>13682</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>IAMV.BNC1KV.CLUSTER</td>
<td>252</td>
<td>16429455</td>
<td>VSAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC1KV.DATA</td>
<td>252</td>
<td>13032467</td>
<td>VSAM</td>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td>0</td>
<td>8192</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC1KV.INDEX</td>
<td>252</td>
<td>3396988</td>
<td>VSAM</td>
<td></td>
<td></td>
<td>2041</td>
<td>8</td>
<td>0</td>
<td>2048</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC125I.CLUSTER</td>
<td>315</td>
<td>359822</td>
<td>IAM</td>
<td>VO</td>
<td>125</td>
<td>125</td>
<td>8</td>
<td>0</td>
<td>13682</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>IAMV.BNC125V.CLUSTER</td>
<td>279</td>
<td>8135073</td>
<td>VSAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC125V.DATA</td>
<td>279</td>
<td>5657533</td>
<td>VSAM</td>
<td></td>
<td>125</td>
<td>8</td>
<td>0</td>
<td>13682</td>
<td>10</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC125V.INDEX</td>
<td>279</td>
<td>2477540</td>
<td>VSAM</td>
<td></td>
<td>2041</td>
<td>8</td>
<td>0</td>
<td>2048</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC250I.CLUSTER</td>
<td>360</td>
<td>777792</td>
<td>IAM</td>
<td>VO</td>
<td>250</td>
<td>250</td>
<td>8</td>
<td>0</td>
<td>13682</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>IAMV.BNC250V.CLUSTER</td>
<td>288</td>
<td>1023859</td>
<td>VSAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC250V.DATA</td>
<td>288</td>
<td>7187432</td>
<td>VSAM</td>
<td></td>
<td>250</td>
<td>8</td>
<td>0</td>
<td>8192</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC250V.INDEX</td>
<td>288</td>
<td>3051160</td>
<td>VSAM</td>
<td></td>
<td>2041</td>
<td>8</td>
<td>0</td>
<td>2048</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC500I.CLUSTER</td>
<td>374</td>
<td>1659165</td>
<td>IAM</td>
<td>VO</td>
<td>500</td>
<td>500</td>
<td>8</td>
<td>0</td>
<td>13682</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>IAMV.BNC500V.CLUSTER</td>
<td>288</td>
<td>12599784</td>
<td>VSAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC500V.DATA</td>
<td>288</td>
<td>9237824</td>
<td>VSAM</td>
<td></td>
<td>500</td>
<td>8</td>
<td>0</td>
<td>8192</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC500V.INDEX</td>
<td>288</td>
<td>3361960</td>
<td>VSAM</td>
<td></td>
<td>2041</td>
<td>8</td>
<td>0</td>
<td>2048</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 40-9: Sample Data Set Summary Report*
### Data Set Summary Report Fields

The fields that are unique to the Data Set Summary Report are described below. The Data Set Name, Total Use Count, and Total EXCP's are identical to the fields in the EXCP Report. For uncataloged VSAM datasets there will be no information for the fields RECFM, AVG LRECL, RKP, or FREE SPACE.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Set Name</strong></td>
<td>Specifies the name of the data set. For cataloged VSAM data sets, this will have the cluster name on the first line, followed by the component names on subsequent lines, which are indented.</td>
</tr>
<tr>
<td><strong>Use Count</strong></td>
<td>Indicates the number of SMF records processed for this data set. This will usually be the number of times the data set was closed during the time interval of the report.</td>
</tr>
<tr>
<td><strong>Total EXCPs</strong></td>
<td>The accumulated total EXCP counts from all of the SMF records processed for this data set. For cataloged VSAM clusters, this number will be the total for all of the components, with the component value given on the subsequent lines.</td>
</tr>
<tr>
<td><strong>DSORG</strong></td>
<td>Indicates the data set organization for the file.</td>
</tr>
<tr>
<td><strong>RECFM</strong></td>
<td>Indicates the record format. The following values are possible:</td>
</tr>
<tr>
<td></td>
<td><strong>KSDS</strong> – For a cataloged VSAM Cluster, indicates that it is a VSAM KSDS.</td>
</tr>
<tr>
<td></td>
<td><strong>ESDS</strong> – For a cataloged VSAM Cluster, indicates that it is a VSAM ESDS.</td>
</tr>
<tr>
<td></td>
<td><strong>RRDS</strong> – For a cataloged VSAM Cluster, indicates that it is a VSAM RRDS.</td>
</tr>
<tr>
<td></td>
<td><strong>VRRDS</strong> – For a cataloged VSAM Cluster, indicates that it is a VSAM Variable RRDS.</td>
</tr>
<tr>
<td></td>
<td><strong>F[B]</strong> - Fixed length record file. For IAM, can only be a Compatible Format data sets. For VSAM, indicates that file was defined with an equal average and maximum record length.</td>
</tr>
<tr>
<td></td>
<td><strong>VB</strong> - Variable length record file, applicable to VSAM or Compatible Format IAM data sets.</td>
</tr>
<tr>
<td></td>
<td><strong>VE</strong> - Variable length record, IAM Enhanced Format file.</td>
</tr>
<tr>
<td></td>
<td><strong>VO</strong> - Variable length record IAM Enhanced Format file with Variable Overflow enabled.</td>
</tr>
<tr>
<td></td>
<td><strong>-DC</strong> - IAM Data Compressed file.</td>
</tr>
<tr>
<td><strong>AVG LRECL</strong></td>
<td>Indicates the defined average record length.</td>
</tr>
<tr>
<td><strong>MAX LRECL</strong></td>
<td>Indicates the maximum defined logical record length.</td>
</tr>
<tr>
<td><strong>KEY LENGTH</strong></td>
<td>Indicates the defined key length for the data set.</td>
</tr>
<tr>
<td><strong>RKP</strong></td>
<td>Indicates the defined Relative Key Position (or key offset) for the data set.</td>
</tr>
<tr>
<td><strong>BLK or CI SIZE</strong></td>
<td>Indicates the VSAM CI Size or the IAM block size of the data set.</td>
</tr>
<tr>
<td><strong>FREESPACE</strong></td>
<td>Indicates the defined values for CI and CA percent freespace.</td>
</tr>
</tbody>
</table>
IAMSMFVS Size Reports

After the Data Set Summary Report, IAMSMFVS produces size reports. Like the EXCP reports, the Size reports are produced by data set organization, with only the largest data sets included in each report. The number of data sets included in each size report is from the keyword MAXREPORTS, which defaults to 100. If VSAM data sets are being reported on, that VSAM Size Report will appear first. The IAM Size report, if IAM data sets are being reported on, appears after the VSAM Size Report.

VSAM Size Report

<table>
<thead>
<tr>
<th>DATA SET NAME</th>
<th>ALLOC TRKS</th>
<th>TOTAL EXCPS</th>
<th>USE COUNT</th>
<th>EXTENTS</th>
<th>AVG LRECL</th>
<th>MAX LRECL</th>
<th>KEY LEN</th>
<th>RKP</th>
<th>CISIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAMV.BNC1KV.CLUSTER</td>
<td>30195</td>
<td>16429455</td>
<td>252</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC1KV.DATA</td>
<td>30060</td>
<td>13032467</td>
<td>252</td>
<td>1</td>
<td>1000</td>
<td>8</td>
<td>0</td>
<td>8192</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC1KV.INDEX</td>
<td>135</td>
<td>3396988</td>
<td>252</td>
<td>1</td>
<td>2041</td>
<td>8</td>
<td>0</td>
<td>2048</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC500V.CLUSTER</td>
<td>25185</td>
<td>12599784</td>
<td>288</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC500V.DATA</td>
<td>25050</td>
<td>9237824</td>
<td>288</td>
<td>1</td>
<td>500</td>
<td>8</td>
<td>0</td>
<td>8192</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC500V.INDEX</td>
<td>135</td>
<td>3361960</td>
<td>288</td>
<td>1</td>
<td>2041</td>
<td>8</td>
<td>0</td>
<td>2048</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC750V.CLUSTER</td>
<td>20175</td>
<td>16481568</td>
<td>288</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC750V.DATA</td>
<td>20040</td>
<td>12789824</td>
<td>288</td>
<td>1</td>
<td>750</td>
<td>8</td>
<td>0</td>
<td>8192</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC750V.INDEX</td>
<td>135</td>
<td>3691744</td>
<td>288</td>
<td>1</td>
<td>2041</td>
<td>8</td>
<td>0</td>
<td>2048</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC250V.CLUSTER</td>
<td>12660</td>
<td>10238592</td>
<td>288</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC250V.DATA</td>
<td>12525</td>
<td>7187432</td>
<td>288</td>
<td>1</td>
<td>250</td>
<td>8</td>
<td>0</td>
<td>8192</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC250V.INDEX</td>
<td>135</td>
<td>3051160</td>
<td>288</td>
<td>1</td>
<td>2041</td>
<td>8</td>
<td>0</td>
<td>2048</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC125V.CLUSTER</td>
<td>5145</td>
<td>8135073</td>
<td>279</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC125V.DATA</td>
<td>5010</td>
<td>5657533</td>
<td>279</td>
<td>1</td>
<td>125</td>
<td>8</td>
<td>0</td>
<td>8192</td>
<td></td>
</tr>
<tr>
<td>IAMV.BNC125V.INDEX</td>
<td>135</td>
<td>2477540</td>
<td>279</td>
<td>1</td>
<td>2041</td>
<td>8</td>
<td>0</td>
<td>2048</td>
<td></td>
</tr>
</tbody>
</table>

IAM Size Report

<table>
<thead>
<tr>
<th>DATA SET NAME</th>
<th>TRACKS USED</th>
<th>TOTAL IAM SIZE REPORT</th>
<th>TOTAL RECORDS</th>
<th>INDEPENDENT OVERFLOW %</th>
<th>PRIME EXT</th>
<th>CI%</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAMV.BNC1KI.CLUSTER</td>
<td>41438</td>
<td>4074098</td>
<td>343</td>
<td>2000000</td>
<td>43069</td>
<td>20051</td>
</tr>
<tr>
<td>IAMV.BNC750I.CLUSTER</td>
<td>31009</td>
<td>2922216</td>
<td>391</td>
<td>2000000</td>
<td>44694</td>
<td>15808</td>
</tr>
<tr>
<td>IAMV.BNC500I.CLUSTER</td>
<td>20664</td>
<td>1659165</td>
<td>374</td>
<td>2000000</td>
<td>44658</td>
<td>14828</td>
</tr>
<tr>
<td>IAMV.BNC250I.CLUSTER</td>
<td>10335</td>
<td>777792</td>
<td>360</td>
<td>2000000</td>
<td>43725</td>
<td>15055</td>
</tr>
<tr>
<td>IAMV.BNC125I.CLUSTER</td>
<td>5218</td>
<td>398822</td>
<td>315</td>
<td>2000000</td>
<td>45898</td>
<td>13032</td>
</tr>
</tbody>
</table>

Figure 40-11: Example IAMSMFVS IAM Size Report
### Size Report Field Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Set Name</td>
<td>Specifies the name of the data set. For cataloged VSAM data sets, this will have the cluster name on the first line, followed by the component names on subsequent lines, which are indented. For uncataloged VSAM data sets, only the component name will be printed.</td>
</tr>
<tr>
<td>Alloc Tracks</td>
<td>For VSAM components, the maximum number of tracks allocated to the component.</td>
</tr>
<tr>
<td>Tracks Used</td>
<td>For IAM data sets, indicates the maximum amount of DASD space used for this file.</td>
</tr>
<tr>
<td>Total EXCPS</td>
<td>The accumulated total EXCP counts from all of the SMF records processed for this data set. For cataloged VSAM clusters, this number will be the total for all of the components, with the component value given on the subsequent lines.</td>
</tr>
<tr>
<td>Use Count</td>
<td>Indicates the number of SMF records processed for this data set. This will usually be the number of times the data set was closed during the time interval of the report.</td>
</tr>
<tr>
<td>Extents</td>
<td>For VSAM clusters, indicates maximum number of DASD extents.</td>
</tr>
<tr>
<td>AVG LRECL</td>
<td>For VSAM data components, indicates the defined average record length.</td>
</tr>
<tr>
<td>MAX LRECL</td>
<td>For VSAM, indicates the maximum defined logical record length.</td>
</tr>
<tr>
<td>KEY LENGTH</td>
<td>For VSAM, indicates the defined key length for the data set.</td>
</tr>
<tr>
<td>RKP</td>
<td>For VSAM, indicates the defined Relative Key Position (or key offset) for the data set.</td>
</tr>
<tr>
<td>CI SIZE</td>
<td>For VSAM, indicates the VSAM CI Size which is the amount of data transferred per physical I/O.</td>
</tr>
<tr>
<td>Total Records</td>
<td>For IAM, contains the maximum records that were in the file, or if CURRENT was specified, the current number of records in the file.</td>
</tr>
<tr>
<td>Max Rec</td>
<td>For Enhanced Format files, without an overflow override, the estimated number of records that can fit within the currently allocated extents.</td>
</tr>
<tr>
<td>Use Rec</td>
<td>For Enhanced Format files with an Overflow override, this will be that value.</td>
</tr>
<tr>
<td>% USE</td>
<td>For Compatible format files, the maximum size of the overflow area.</td>
</tr>
<tr>
<td>Prime Ext</td>
<td>The maximum records in overflow, or if CURRENT was specified on the REPORT card, in which case it is the most recent value.</td>
</tr>
<tr>
<td>CI%</td>
<td>The percentage of the overflow area that has been used.</td>
</tr>
<tr>
<td></td>
<td>For IAM data sets, the size of Integrated Overflow, normally specified by CI Freespace.</td>
</tr>
</tbody>
</table>
41 SMF BASIC REPORTING UTILITY

41.01 IAMSMF Overview

Overview
The IAM product includes a specialized SMF reporting program called IAMSMF. This program is intended to aid customers in converting VSAM data sets to IAM and in using IAM data sets in their installation. Customers can use IAMSMF to:

1. Aid in conversion of data sets from VSAM to IAM, by providing the capability to determine what jobs and job steps are defining the VSAM data sets, as well as processing the selected data sets.
2. Provide reports on SMF job statistics to help evaluate the benefit of IAM. You can obtain reports that include CPU time, EXCP's, and elapsed time for jobs using VSAM data sets. Then, compare the jobs based on those statistics to see the improvements realized after converting the VSAM data sets to IAM.
3. Provide detailed reports on IAM data set activity by producing the IAMINFO reports from the IAM SMF records.
4. Find out what jobs or users are accessing data sets.
5. Print or copy selected SMF records.

Functional Summary
The IAMSMF program has been designed to give users the ability to extract data and statistics from Systems Management Facility (SMF) records. The functions include data set query, printing IAMINFO reports, job reporting, record copy, and record print. IAMSMF is a batch program, which based upon simple control statements, will extract resource utilization information from SMF job/step and data set records. Information can be selected directly from the active system SMF data sets, from SMF format archival files, or from extract files created by the COPY function of the program itself.

Command Summary
IAMSMF has the following commands:

- **COPY**: Copy selected SMF records to a sequential file creating a subset of the data for later use.
- **IAMINFO**: Produce IAMINFO reports from IAM generated SMF records.
- **PRINT**: Print selected SMF records in dump (Hexadecimal) format.
- **QUERY**: Produces a data set oriented report, to find out what job steps use the specified data sets, and how they are used. Can be requested for specific data set name(s) or by data set name prefixes (data set groups).
- **REPORT**: Produces a job oriented report, with a break out or resource and data set utilization by job step. The report is produced in chronological order from the available SMF data. Can be requested for a specific Jobname(s) or by Job Group Name(s).

**WARNING**
Note: The REPORT process is provided on an as-is basis. The use of the reports generated by the IAMSMF REPORT process are intended for general informational purposes provided to help evaluate the IAM product. The results are not certified to meet with all of the complexities involved in processing SMF data which are beyond the scope of the intentions of this program.
41.02 IAMSMF JCL Requirements

The JCL statements required to execute IAMSMF are as follows:

**Execute Statement**

Specifies the name of the IAM SMF Analysis program: IAMSMF. The region size used by IAMSMF is at least 1024K. A typical execute statement would be:

```
//SMFQUERY EXEC PGM=IAMSMF,REGION=1M
```

**DD Statements**

The following table identifies the required DD statements.

<table>
<thead>
<tr>
<th>DD Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB or JOBLIB</td>
<td>It is recommended that the IAM program load library be included in the system link list. If it is not you must include a STEPLIB/JOBLIB DD statement specifying the IAM load library that contains the IAMSMF program.</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>Specifies where the IAMSMF control statements, messages and reports are to be printed. Usually a SYSOUT data set.</td>
</tr>
<tr>
<td>SYSMF</td>
<td>Specifies the SMF source data set. This DD statement may point to one or more of the active SMF data sets on disk (ex: SYS1.MANx), an SMF format archival file, or a sequential file produced by the COPY function of this program. An alternate DD name may be specified via a control statement.</td>
</tr>
<tr>
<td>SYSUT2</td>
<td>Specifies the output data set for a COPY operation. Usually a sequential file on tape or disk. An alternate DD name may be specified via a control statement.</td>
</tr>
<tr>
<td>SYSIN</td>
<td>Specifies the input control statement card image data set. Usually a DD * data set.</td>
</tr>
<tr>
<td>IAMCSV</td>
<td>If allocated, will cause IAMINFO output to be generated as Comma Separated Values (CSV) records which can be imported into programs such as Excel to view IAMINFO report data as a spreadsheet. This should be a PS file with no other DCB attributes specified.</td>
</tr>
</tbody>
</table>

**Example A: Basic IAMSMF JCL Example**

An example of the JCL to run IAMSMF is shown below. The SYSUT2 DD card is required for a COPY command, otherwise it is not needed.

```
//IAMSMF   EXEC PGM=IAMSMF,REGION=1M
//STEPLIB  DD   DISP=SHR,DSN=iam.load.lib
//SYSPRINT DD   SYSOUT=*
//SYSMF    DD   DISP=SHR,DSN=smf.data.set
//SYSUT2   DD   DSN=my.smf.records,UNIT=SYSDA,
//           DISP=(,CATLG),,SPACE=(CYL,(10,1))
//SYSIN    DD   *

IAMSMF Control cards are inserted here

/*
```

*Figure 41-1: Example of Basic JCL to run IAMSMF (EX4102A)*
41.03 IAMSMF - COPY Command

COPY Command
The COPY command is used to copy selected SMF records to a sequential file from either a history file (RECFM=VBS) or an active SMF data recording file. The COPY command can be used to create a subset of the full SMF data so that reports can be generated at a later point in time for comparisons. You might want to save a subset of the records, such as all of the IAM SMF records, in one place and/or for a longer period of time than the normal SMF data is kept. Another use of copy is to aid in creating reports to find out particular information. For example, if you want to find all the jobs that are defining VSAM or IAM data sets, but did not want them intermixed with other file activity, first copy the record types 61 and 63, which are the VSAM Define SMF records. Then run the IAMSMF Query command from the data subset.

Figure 41-2: IAMSMF COPY Command Operands


<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLRECORDS</td>
<td>Specifies that the SMF record type does not participate in record selection.</td>
</tr>
<tr>
<td></td>
<td>The default is deferred to the operand RECTYPE.</td>
</tr>
<tr>
<td></td>
<td>NOTE: This operand conflicts with the operand RECTYPE.</td>
</tr>
<tr>
<td>CHECKLENGTH</td>
<td>Specifies that SMF data records are to be validated against a table of minimum record lengths. If the user has modified the minimum record length or is executing in a non-compatible system, the correct record lengths may be specified by the RECSIZE operand.</td>
</tr>
<tr>
<td></td>
<td>The default is SMF record length is not validated.</td>
</tr>
<tr>
<td>ERRORPRINT</td>
<td>Specifies that any SMF data record that causes an error during processing or fails length verification is to be printed.</td>
</tr>
<tr>
<td></td>
<td>The default is records in error are not printed.</td>
</tr>
</tbody>
</table>
**Operand** | **Description**  
--- | ---  
FROMDATE= | Specifies the lower date limit of the SMF records that are to be copied. The date has the format of yyyyddd or yyddd. The shorter format assumes a prefix of 19.  
| The default, if the FROMDATE and/or TODATE operands are not specified, is that the date of the SMF record will not participate in the selection criteria.  
FROMDDNAME= | Specifies the DDNAME of the SMF file to be used as input to IAMSMF. The default input DDNAME is SYSMF.  
GROUPNAMES= | Specifies that only records having a jobname which begin with the specified character string(s) will be copied. This operand specifies a partial jobname from 1 to 8 characters in length. Up to 50 job groups and/or names may be specified for a single command if entered as follows:  
GROUPNAMES=(jobname1,...,jobnamex)  
| The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.  
IAMRECORDS | Specifies that SMF records generated by IAM are to be selected.  
| The default is deferred to the operand RECTYPE.  
JOBNAMES= | Specifies that only records having a jobname which match the jobname(s) specified will be copied. This operand specifies a complete jobname from 1 to 8 characters in length. Up to 50 job names may be specified in a single command if entered as follows:  
JOBNAMES=(jobname1,...,jobnamex)  
| The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.  
MAXRECLENGTH= H= | Specifies the largest SMF record that the program will process. The number may be any value from 16384 to 65536, inclusive.  
| The default is 16384 bytes.  
PRTLENGTH= | Limits the amount of data to be printed if the ERRORPRINT keyword is specified. The number may be any value from 32 to 65536, inclusive.  
| The default is 32768 bytes.  
RECSIZE(rrr)= | Establishes the minimum length of the SMF record type ‘rrr’ as the value ‘nnnn’.  
| The default minimum record lengths for system generated SMF records are documented in the IBM Systems Management Facilities manual.
<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECTYPE=</td>
<td>Identifies the specific record type(s) to be copied. Up to 50 record types may be specified for a single command if entered as follows: RECTYPE=(rrr,...,rrr)</td>
</tr>
<tr>
<td></td>
<td>The SMF record types which will be copied by default (if ALLRECORDS is not specified) are as follows:</td>
</tr>
</tbody>
</table>
|           | 4 - Step termination  
|           | 5 - Job termination  
|           | 14 - NON-VSAM Dataset CLOSED (input)  
|           | 15 - NON-VSAM Dataset CLOSED (output/update)  
|           | 20 - Job initiation  
|           | 30 - Common Address Space Work Record  
|           | 34 - TSO session termination  
|           | 64 - VSAM Dataset CLOSED  |
|           | NOTE: This operand conflicts with the operand ALLRECORDS. |
| TODATE=   | Specifies the upper date limit of the SMF records that are to be copied. The format is yyyyddd or yyddd. The shorter format assumes a prefix of 19. |
|           | The default, if the FROMDATE and/or TODATE operands are not specified, is that the date of the SMF record will not participate in the selection criteria. |
| TODDNAME= | Specifies the DDNAME of the output file for the copied SMF records. |
|           | The default output DDNAME is SYSUT2. |
This example shows how to create a subset of SMF records that can be used for subsequent input to the IAMSMFVS analysis program. The record types being copied include the Type 30 for the step termination, the Type 64 for VSAM data sets, and the Type 201, which is the IAM SMF record type specified in the IAM Global Options Table. The output data set created by this job can be used as an input data set for IAMSMFVS. The FROMDATE and TODATE keywords are specified indicating to copy data for the month of November 2014. The SYSUT2 DD specifies a DISP=MOD, indicating that records are being added to an already existing file, which contains previously extracted data from prior months. A MAXRECLENGTH keyword was specified to insure that there would be no records dropped.

```
//IAMS MF   EXEC PGM=IAMS MF,REGION=1M
//ST EPLIB  DD   DISP=SHR,DSN=iam.load.lib
//SYSPRINT DD   SYSOUT=* 
//SYSMF    DD   DISP=SHR,DSN=smf.data.set
//SYSUT2   DD   DSN=my.smf.records,DISP=(MOD,KEEP,KEEP)
//SYSIN    DD    */
COPY      RECTYPE=(30,64,201),FROMDATE=2014.305,
          TODATE=2014.334,MAXRECLENGTH=32768
/*
```

Figure 41-3: Example of IAMSMF COPY Command (EX4103A)
**IAMINFO Command**

The IAMINFO command is used to produce the detailed IAMINFO reports from IAM generated SMF records. IAM will normally produce an IAMINFO report each time an IAM data set is closed if the job step has an IAMINFO DD card. By collecting the IAM SMF records, you can still get the IAMINFO reports without changing the JCL in every job using IAM data sets. Plus, this offers the advantage of being able to obtain the IAMINFO reports only on an as needed or desired basis, or if the original job output has been discarded.

To obtain a summarization of IAM data set activity, use the IAMSMFVS program. That program provides three distinct reports, consisting of one line per data set rather than multiple full page reports. You could use IAMSMFVS to obtain the summary information, and then use the IAMSMF IAMINFO command to get detailed information about specific files.

Each IAMINFO report is one page in length, providing detailed information about the data set and its use. An IAMINFO report is generated by IAMSMF as one per data set per job step. Optionally, IAMINFO reports can be printed by IAMSMF for each time the data set is closed, or once per data set per job. From the IAMINFO reports, you can tell how a program is accessing the data set, whether or not more buffers could help improve performance, and if the data set needs to be reorganized.

If you would prefer to view the IAMINFO output as a spreadsheet add an IAMCSV DD, described above, to your JCL and IAMSMF will generate one line for each IAMINFO report in a Comma Separated Values (CSV) format which can be imported in to any spreadsheet program.

The IAMINFO command offers a wide variety of selection criteria to provide a lot of flexibility in the reports that end up being printed.

```
IAMINFO
[ATTRIBUTE=cccccc] [,MAXSTACK=nnnnn]
[,BLOCKS=nnnnnn] [,MERGE=cccccc]
[,CYLS=nnnnn] [,NUMCMD=nnnnn]
[,DGROUPS=dname] [,NUMRECS=nnnnn]
[,DSNAMES=dname] [,OFULL=nn]
[,DYRET=nnnnn] [,ORECS=nnnnn]
[,EXCP=nnnnn] [,PEBLKS=nnnnn]
[,FROMDATE=yyyyddd] [,PEFULL=nn]
[,FROMDDNAME=ddname] [,RECTYPE=nnn]
[,GROUPNAMES=jobname] [,TODATE=yyyyddd]
[,JOBNAMES=jobname] [,TRACKS=nnnnn]
[,LRECL=nnnnn] [,TYPECMD[E]=(a,b,c)]
[,SHAREOPTION=n]
```

*Figure 41-4: IAMSMF IAMINFO Command Operands*
<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE=</td>
<td>Identifies which attributes an IAM file must have to participate in record selection. Valid values are:</td>
</tr>
<tr>
<td>DATACOMP</td>
<td>The IAM file contains compressed data.</td>
</tr>
<tr>
<td>KEYCOMP</td>
<td>The IAM file contains compressed keys.</td>
</tr>
<tr>
<td>INCPEBLK</td>
<td>IAM found that a record that could have gone into PRIME EXTENSION, but was placed into INDEPENDENT OVERFLOW because the PRIME EXTENSION was full. Redefining this file with a larger PRIME EXTENSION may improve performance. An IAM371 message was issued if an IAMINFO DD was present during the job's execution.</td>
</tr>
<tr>
<td>MOREBUFFER</td>
<td>IAM found that additional buffers could have been used to improve performance, however the MAXBUF or BUFNO option prevented IAM from acquiring more buffers. An IAM368 message was issued if an IAMINFO DD was present during the job's execution.</td>
</tr>
<tr>
<td>NOCORE</td>
<td>There was not enough storage available in the region for IAM to acquire additional buffers. An IAM367 message was issued if an IAMINFO DD was present during the job's execution.</td>
</tr>
<tr>
<td>REORG</td>
<td>IAM files that have received warning messages recommending that the file be reorganized.</td>
</tr>
<tr>
<td></td>
<td>By default, the data set attributes do not participate in the selection criteria.</td>
</tr>
<tr>
<td></td>
<td>This operand supports the following logical operators: =, ¬=.</td>
</tr>
<tr>
<td>BLOCKS=</td>
<td>Specifies the size, in blocks, of the files to be selected.</td>
</tr>
<tr>
<td></td>
<td>By default, the size of the IAM data set does not participate in the selection criteria.</td>
</tr>
<tr>
<td></td>
<td>This operand supports the following logical operators: =, ¬=, &gt;, &gt; =, &lt;, &lt; =</td>
</tr>
<tr>
<td>CYLS=</td>
<td>Specifies the size, in cylinders, of the files to be selected.</td>
</tr>
<tr>
<td></td>
<td>By default, the size of the IAM data set does not participate in the selection criteria.</td>
</tr>
<tr>
<td></td>
<td>This operand supports the following logical operators: =, ¬=, &gt;, &gt; =, &lt;, &lt; =</td>
</tr>
<tr>
<td>DSGROUPS=</td>
<td>Specifies that only records having a data set name which begin with the given character string(s) will be processed. This operand specifies a partial dataset name from 1 to 44 characters in length. Up to 50 data set groups may be specified for a single command if entered as follows: DSGROUPS=(dsname1,...,dsnamex) By default, the name of the IAM data set does not participate in the selection criteria.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Operand</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
</tr>
</tbody>
</table>
| **DSNAMES=** | Specifies that only records having a data set name which match the dataset name(s) specified will be processed. This operand specifies a complete data set name from 1 to 44 characters in length. Up to 50 data set names may be specified for a single command if entered as follows:  

\[
\text{DSNAMES}=(\text{dsname},...,\text{dsname})
\]  

By default, the name of the IAM data set does not participate in the selection criteria. |
| **DYNRET=** | Establishes the limit of record retrievals from IAM's Dynamic Table.  

By default, the number of dynamic retrievals does not participate in the selection criteria.  

This operand supports the following logical operators: =, ¬ =, >, ≥, <, ≤ |
| **EXCP=** | Establishes the limit of EXCPs (physical read and writes) to the IAM file.  

By default, the number of EXCPs does not participate in the selection criteria.  

This operand supports the following logical operators: =, ¬ =, >, ≥, <, ≤ |
| **FROMDATE=** | Specifies the lower date limit of the SMF records that are to be copied.  

The default, if the FROMDATE and/or TODATE operands are not specified, is that the date of the SMF record will not participate in the selection criteria. |
| **FROMDNAME=** | Specifies the DDNAME of the SMF file to be used as input to IAMSMF.  

The default input DDNAME is SYSMF. |
| **GROUPNAMES=** | Specifies that only those records having a job name which begin with the specified character string(s) will be selected. This operand specifies a partial job name from 1 to 8 characters in length. Up to 50 job groups and/or names may be specified for a single command if entered as follows:  

\[
\text{GROUPNAMES}=(\text{jobname1},...,\text{jobnamex})
\]  

The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the job name will not participate in SMF record selection. |
| **JOBNAMES=** | Specifies that only records having a job name which match the jobname(s) specified will be selected. This operand specifies a complete job name from 1 to 8 characters in length. Up to 50 job names may be specified in a single command if entered as follows:  

\[
\text{JOBNAMES}=(\text{jobname1},...,\text{jobnamex})
\]  

The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the job name will not participate in SMF record selection. |
IAMSMF - IAMINFO Command

<table>
<thead>
<tr>
<th><strong>Operand</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>LRECL=</td>
<td>Establishes the limit for the length of records (maximum length for variable) within an IAM file. By default, the record length does not participate in the selection criteria. This operand supports the following logical operators: =, ≠, &gt;, ≥, &lt;, ≤</td>
</tr>
<tr>
<td>MAXSTACK=</td>
<td>Specifies the maximum number of compressed SMF records that can be retained in storage. The number can be any value from 100 to 50000, inclusive. The default is 2500 records.</td>
</tr>
<tr>
<td>MERGE=</td>
<td>Identifies how IAMSMF will process multiple SMF records created for the same dataset by the same job. Valid values are: JOB: All SMF records referencing the same IAM data set within a job will be merged into a single report. NO: No IAM SMF records are to be merged. An IAMINFO report will be generated for each IAM SMF record found. STEP: All SMF records referencing the same IAM data set within a job step (i.e. multiple OPEN/CLOSE) will be merged into a single report. The default is STEP.</td>
</tr>
<tr>
<td>NUMCMTDS=</td>
<td>Establishes the limit of the total number of commands issued against the IAM file. When used in conjunction with the TYPECMD operand, the selection will be limited to those files having a command count of that type. By default, the number of commands issued against the IAM file does not participate in the selection criteria. This operand supports the following logical operators: =, ≠, &gt;, ≥, &lt;, ≤</td>
</tr>
<tr>
<td>OFULL=</td>
<td>Establishes the percent of overflow used limit within an IAM file. By default, the percent of overflow used does not participate in the selection criteria. This operand supports the following logical operators: =, ≠, &gt;, ≥, &lt;, ≤</td>
</tr>
<tr>
<td>ORECS=</td>
<td>Establishes the number of overflow records limit within an IAM file. By default, the number of overflow records does not participate in the selection criteria.</td>
</tr>
<tr>
<td>PEBLKS=</td>
<td>Establishes the number of prime extension blocks limit within an IAM file. By default, the number of prime extension blocks does not participate in the selection criteria. This operand supports the following logical operators: =, ≠, &gt;, ≥, &lt;, ≤</td>
</tr>
<tr>
<td><strong>Operand</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PEFULL=</td>
<td>Establishes the percent of prime extension blocks used limit within an IAM file.</td>
</tr>
<tr>
<td></td>
<td>By default, the percent of prime extension blocks used does not participate in</td>
</tr>
<tr>
<td></td>
<td>the selection criteria.</td>
</tr>
<tr>
<td></td>
<td>This operand supports the following logical operators: =, ¬ =, &gt;, &gt; =, &lt;, &lt; =</td>
</tr>
<tr>
<td>RECTYPE=</td>
<td>Identifies the record type of the IAM generated SMF record. The default record</td>
</tr>
<tr>
<td></td>
<td>type is that which is defined in the IAM option table. Refer to Section 91 for</td>
</tr>
<tr>
<td></td>
<td>further documentation on the IAM option table.</td>
</tr>
<tr>
<td>SHAREOPTION=</td>
<td>Specifies the cross region share options of the files to be selected. Valid</td>
</tr>
<tr>
<td></td>
<td>values are 1, 2, 3, or 4.</td>
</tr>
<tr>
<td></td>
<td>By default, the share options do not participate in the selection criteria.</td>
</tr>
<tr>
<td></td>
<td>This operand supports the following logical operators: =, ¬ =, &gt;, &gt; =, &lt;, &lt; =</td>
</tr>
<tr>
<td>TODATE=</td>
<td>Specifies the upper date limit of the SMF records that are to be copied.</td>
</tr>
<tr>
<td></td>
<td>The default, if the FROMDATE and/or TODATE operands are not specified, is that</td>
</tr>
<tr>
<td></td>
<td>the date of the SMF record will not participate in the selection criteria.</td>
</tr>
<tr>
<td>TRACKS=</td>
<td>Specifies the size, in tracks, of the files to be selected.</td>
</tr>
<tr>
<td></td>
<td>By default, the size of the IAM data set does not participate in the selection</td>
</tr>
<tr>
<td></td>
<td>criteria.</td>
</tr>
<tr>
<td></td>
<td>This operand supports the following logical operators: =, ¬ =, &gt;, &gt; =, &lt;, &lt; =</td>
</tr>
<tr>
<td>TYPECMDE=</td>
<td>Identifies the type of command(s) that must have been issued against Enhanced</td>
</tr>
<tr>
<td></td>
<td>Format IAM files.</td>
</tr>
<tr>
<td></td>
<td>If NUMCMDS is also specified, the SMF record will be selected if that number</td>
</tr>
<tr>
<td></td>
<td>of commands of that type was issued.</td>
</tr>
<tr>
<td></td>
<td>If NUMCMDS is not specified, the SMF record will be selected if ANY command of</td>
</tr>
<tr>
<td></td>
<td>that type was issued.</td>
</tr>
<tr>
<td></td>
<td>Up to 16 command types may be specified if entered as follows:</td>
</tr>
<tr>
<td></td>
<td>TYPECMDE=(c,c,...,c)</td>
</tr>
<tr>
<td></td>
<td>The following values (command types which correspond to those generated by</td>
</tr>
<tr>
<td></td>
<td>IAMINFO) are supported:</td>
</tr>
<tr>
<td>A - Add, C - Close, E - Erase, F - Skip Sequential Get, G - Get Sequential, I -</td>
<td></td>
</tr>
<tr>
<td>IAM Statistics, K - Point, L - Record Length Change, N - Point Key Greater or</td>
<td></td>
</tr>
<tr>
<td>Equal, or Generic, O - Open, P - Get Previous (Backwards Get), R - Random Read</td>
<td></td>
</tr>
<tr>
<td>with Key Equal, S - Random Read with Key Greater or Equal or Generic, T - Close</td>
<td></td>
</tr>
<tr>
<td>Type=T, U - Put/ Write for Update, V - Verify, or X - Buffer Flush Request.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>By default, the type of commands issued against the IAM file does not</td>
</tr>
<tr>
<td></td>
<td>participate in the selection criteria.</td>
</tr>
</tbody>
</table>
IAMINFO Output  
The IAMINFO output produced by IAMSMF is quite similar to the normal IAMINFO report. The only change has been in the heading messages. When produced by IAMSMF, the IAMINFO reports do not have the IAM360 or IAM361 heading messages. Rather, they have an IAM370 JOB CHARACTERISTICS section. A sample of that portion of the IAMINFO report is shown below. For a complete description of the IAMINFO report, please refer to Section 10.72 IAMINFO Reports.

IAM370  JOB CHARACTERISTICS -
JOB NAME --------------- =  RAM270  -  STEP NAME ----------------==  KSD2708
PROGRAM NAME ----------- =  IAMTVSAM  -  FUNCTION --------------- =  UPDATE PROCESSING
DDNAME ---------------- =  VSAMCRT1  -  DSNAME = IAMV.KSD270.CLUSTER
DATE OPENED ------------ =  2014.295  -  TIME OPENED ----------- =  07:51:34
DATE CLOSED ----------- =  2014.295  -  TIME CLOSED ----------- =  07:51:34

Figure 41-5: Sample of IAM370 IAMINFO Output

Job Characteristics Field Description

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description of Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB NAME</td>
<td>Indicates the name of the job that has processed this IAM data set.</td>
</tr>
<tr>
<td>STEP NAME</td>
<td>Indicates the name of the job step that has processed this IAM data set.</td>
</tr>
<tr>
<td>PROGRAM NAME</td>
<td>The name of the program that issued the OPEN for the IAM data set.</td>
</tr>
<tr>
<td>FUNCTION</td>
<td>The function being performed. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• FILE CREATION - Indicates a file load process.</td>
</tr>
<tr>
<td></td>
<td>• UPDATE PROCESSING - Indicates that the file was opened for update.</td>
</tr>
<tr>
<td></td>
<td>• INPUT PROCESSING - Indicates that the file was opened for input processing only.</td>
</tr>
<tr>
<td></td>
<td>• RLS - File is being processed through IAM/PLEX or IAM/RLS</td>
</tr>
<tr>
<td>DDNAME</td>
<td>Indicates the DD name used to OPEN the file.</td>
</tr>
<tr>
<td>DSNAME</td>
<td>Indicates the IAM data set name.</td>
</tr>
<tr>
<td>DATE OPENED</td>
<td>The date that the data set was opened.</td>
</tr>
<tr>
<td>TIME OPENED</td>
<td>The time of the OPEN.</td>
</tr>
<tr>
<td>DATE CLOSED</td>
<td>Indicates the date that the data set was closed.</td>
</tr>
<tr>
<td>TIME CLOSED</td>
<td>Indicates the time of the CLOSE.</td>
</tr>
</tbody>
</table>

Figure 41-6: IAM370 Job Characteristics Field Descriptions
Example A: Requesting a Specific Job and Data Set

In this first example of running IAMINFO from IAMSMF, we are after a specific IAMINFO report. The JOBNAME indicates the name of the job, the DSNAME indicates the name of the data set, and the FROMDATE and TODATE indicate the date that the job was run. MERGE=NO is also specified, so the IAMINFO reports come out for each close, rather than being merged by job step.

```
//IAMSMF EXEC PGM=IAMSMF,REGION=1M
//STEPLIB DD DISP=SHR,DSN=iam.load.lib
//SYSPRINT DD SYSOUT=*  
//SYSMF DD DISP=SHR,DSN=smf.data.set 
//SYsin DD *
  IAMINFO DSNAME=IAMV.KSD270.CLUSTER,JOBNAME=KSD270,
         MERGE=NO,FROMDATE=2014295,TODATE=2014295
/*
```

Figure 41-7: Example 1 of IAMSMF IAMINFO Command (EX4104A)

Example B: IAMINFO Reports for all Data Sets for a Particular Job

This next example is requesting all of the IAMINFO reports for jobs beginning with the specified GROUPNAME that ran on the specified date. This JCL also includes an IAMCSV DD card which means that the IAMINFO data will also be written out in a CSV format.

```
//IAMSMF EXEC PGM=IAMSMF,REGION=1M
//STEPLIB DD DISP=SHR,DSN=iam.load.lib
//SYSPRINT DD SYSOUT=*  
//SYSMF DD DISP=SHR,DSN=smf.data.set  
//IAMCSV DD DSN=my.iaminfo.csv,DISP=(,CATLG),
// SPACE=(CYL,(5,1)),STORCLAS=storclas 
//SYsin DD *
  IAMINFO GROUPNAME=KSD2,MERGE=NO,
     FROMDATE=2014295,TODATE=2014295
/*
```

Figure 41-8: Example 2 of an IAMSMF IAMINFO Command (EX4104B)
Example C: Using Selective Comparisons

In this next example, two IAMINFO commands are being issued. To reduce processing time, the IAM SMF records are first copied to a temporary data set, which is then used for input to the IAMINFO commands. The first IAMINFO is requesting all files opened under the various CICS regions that could have used more buffers to be printed. As part of the screening, we are only concerned about those files that have relatively heavy I/O, so EXCP>10000 are included. From this, a determination can be made if the maximum buffer number really should be raised.

The second IAMINFO command will find jobs that have done a large quantity of inserts. This is done by specifying TYPECMDE=A, which means ADD requests, and NUMCMDS>10000, which indicates more than 10000 adds.

```
//IAMSF   EXEC PGM=IAMSF,REGION=IM
//STEPLIB  DD   DISP=SHR,DSN=iam.load.lib
//SYSPRINT DD   SYSOUT=*  
//SYSMF    DD   DISP=SHR,DSN=smf.data.set
//SYSUT2   DD   UNIT=SYSDA,SPACE=(CYL,(10,5))
//SYSIN   DD    *  
COPY IAMRECORDS
IAMINFO  GROUPNAMES=CICS,ATTR=MOREBUFFER,EXCP>10000,
FROMDDNAME=SYSUT2
IAMINFO  NUMCMDS>10000,TYPECMDE=A,FROMDD=SYSUT2
/*
```

Figure 41-9: Example 3 of IAMSF IAMINFO Command (EX4104C)
41.05 IAMSMF - PRINT Command

The PRINT command is used to print selected SMF records from either a history file (RECFM=VBS) or an active SMF data recording file. The primary use of the PRINT command is typically to diagnose unexpected results from the IAMSMF or IAMSMFVS analysis programs. The PRINT command is useful for verifying that particular types of SMF records are actually being collected, and also to manually verify the format of the record.

```
PRINT
   [ALLRECORDS] [,MAXPRINT= nnn]
   [,CHECKLENGTH] [,MAXRECLENGTH= nnnn ]
   [,ERRORPRINT] [,PRTLENGTH= nnn ]
   [,FROMDATE= yyyyddd ] [,RECSIZE(rrr)= nnnnn ]
   [,FROMDDNAME= ddname] [,RECTYPE= nnn ]
   [,GROUPNAMES= jobname] [,TODATE= yyyyddd ]
   [,JOBNAMES= jobnames ]
```

Figure 41-10: IAMSMF Print Command Operands

<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLRECORDS</td>
<td>Specifies that the SMF record type does not participate in record selection.</td>
</tr>
<tr>
<td></td>
<td>The default is deferred to the operand RECTYPE.</td>
</tr>
<tr>
<td></td>
<td>NOTE: This operand conflicts with the operand RECTYPE.</td>
</tr>
<tr>
<td>CHECKLENGTH</td>
<td>Specifies that SMF data records are to be validated against a table of minimum record lengths. If the user has modified the minimum record length or is executing in a non-compatible system, the correct record lengths may be specified by the RECSIZE operand.</td>
</tr>
<tr>
<td>ERRORPRINT</td>
<td>Specifies that any SMF data record that causes an error during processing or fails length verification is to be printed.</td>
</tr>
<tr>
<td>FROMDATE=</td>
<td>Specifies the lower date limit of the SMF records that are to be printed.</td>
</tr>
<tr>
<td>FROMDDNAME=</td>
<td>Specifies the DDNAME of the SMF file to be used as input to IAMSMF.</td>
</tr>
<tr>
<td></td>
<td>The default input DDNAME is SYSMF.</td>
</tr>
</tbody>
</table>

The default is records in error are not printed.

The default is that there is no lower limit on the dates of the SMF records to eligible to be printed.

The default input DDNAME is SYSMF.
<table>
<thead>
<tr>
<th><strong>Operand</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUPNAMES=</td>
<td>Specifies that only the records having a jobname which begin with the specified character string(s) will be printed. This operand specifies a partial jobname from 1 to 8 characters in length. Up to 50 job groups and/or names may be specified for a single command if entered as follows: GROUPNAMES=(jobname1,...,jobnamex) The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.</td>
</tr>
<tr>
<td>JOBNAMES=</td>
<td>Specifies that only the records having a jobname which match the jobname(s) specified will be printed. This operand specifies a complete jobname from 1 to 8 characters in length. Up to 50 jobnames may be specified in a single command if entered as follows: JOBNAMES=(jobname1,...,jobnamex) The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.</td>
</tr>
<tr>
<td>MAXPRINT=</td>
<td>Specifies the maximum number of records the program will print. The number may be any value from 1 to 65536, inclusive. The default is 20 records.</td>
</tr>
<tr>
<td>MAXRECLENGT=</td>
<td>Specifies the largest SMF record that the program will process. The number may be any value from 16384 to 65536, inclusive. The default is 16384 bytes.</td>
</tr>
<tr>
<td>PRTLENGTH=</td>
<td>Limits the amount of data to be printed if ERRORPRINT is indicated. The number may be any value from 32 to 65536, inclusive. The default is 32768 bytes.</td>
</tr>
<tr>
<td>RECSIZE(rrr)=</td>
<td>Establishes the minimum length of the SMF record type ‘rrr’ as the value ‘nnnn’. The default minimum record lengths for system generated SMF records are documented in the IBM Systems Management Facilities manual.</td>
</tr>
<tr>
<td>RECTYPE=</td>
<td>Identifies the specific record type(s) to be copied. Up to 50 record types may be specified for a single command if entered as follows: RECTYPE=(rrr,....,rrr) The SMF record types which will be copied by default are as follows: 4 - Step termination 5 - Job termination 14 - NON-VSAM Dataset CLOSEd (input) 15 - NON-VSAM Dataset CLOSEd (output/update) 20 - Job initiation 30 - Common Address Space Work Record 34 - TSO session termination 64 - VSAM Dataset CLOSEd</td>
</tr>
</tbody>
</table>

NOTE: This operand can NOT be specified if ALLRECORDS has been specified.
Example A: PRINT

An example of running the IAMSMF PRINT command is shown below, followed by sample output. In this example, to make sure that SMF Type 64 records (VSAM Close) are being produced, the PRINT command specifies RECTYPE=64, and this request is being limited to jobs beginning with a RAM, as specified by the GROUPNAME=RAM.

```
//IAMSMF EXEC PGM=IAMSMF,REGION=1M
//STEPLIB DD DISP=SHR,DSN=iam.load.lib
//SYSPRINT DD SYSOUT=* 
//SYSMF DD DISP=SHR,DSN=smf.data.set 
//SYSIN DD *
PRINT RECTYPE=(64),GROUPNAME=RAM
/*
```

Figure 41-11: Example IAMSMF PRINT Command JCL (EX4105A)

### Table

<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TODATE=</td>
<td>Specifies the upper date limit of the SMF records that are to be printed.</td>
</tr>
</tbody>
</table>

The default, is that the date of the SMF record will not participate in the selection criteria.

Figure 41-12: Sample of PRINT Command Output
IAMSMF - QUERY Command

The QUERY command is used to report on data set activity as requested by data set name or data set group name from a history file (RECFM=VBS), an active SMF data recording file, or from a sequential file produced by the COPY command of the program itself. This command is quite useful for tracking down the jobs and/or TSO users that have been using a data set, and also the jobs that defined and deleted the data set.

**QUERY Command Operands**

<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECKLENGTH</td>
<td>Specifies that SMF data records are to be validated against a table of minimum record lengths. If the user has modified the minimum record length or is executing in a non-compatible system, the correct record lengths may be specified by the RECSIZE operand. The default is that the length of an SMF record is not validated.</td>
</tr>
</tbody>
</table>
| DSGROUPS=        | Specifies that only records having a data set name which begin with the given character string(s) will be processed. This operand specifies a partial dataset name from 1 to 44 characters in length. Up to 50 data set groups may be specified for a single command if entered as follows: `DSGROUPS=(dsname1,...,dsnamex)`  
**NOTE:** If neither the DSGROUPS nor DSNames operand is specified, the QUERY command will be marked in error. |
| DSNames=         | Specifies that only records having a data set name which match the dataset name(s) specified will be processed. This operand specifies a complete data set name from 1 to 44 characters in length. Up to 50 data set names may be specified for a single command if entered as follows: `DSNames=(dsname1,...,dsnamex)`  
**NOTE:** If neither the DSGROUPS nor DSNames operand is specified, the QUERY command will be marked in error. |
<table>
<thead>
<tr>
<th><strong>Operand</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ERRORPRINT</td>
<td>Specifies that any SMF data record that causes an error during processing or fails length verification is to be printed. By default, records in error are not printed.</td>
</tr>
<tr>
<td>FROMDATE</td>
<td>Specifies the lower date limit of the SMF records that are to be selected, in the format of yyyyddd or yyddd. The shorter format assumes a prefix of 19. The default, if the FROMDATE and/or TODATE operands are not specified, is that the date of the SMF record will not participate in the selection criteria.</td>
</tr>
<tr>
<td>FROMDDNAME</td>
<td>Specifies the DDNAME of the SMF file to be used as input to IAMSMF. The default input DDNAME is SYSMF.</td>
</tr>
<tr>
<td>GROUPNAMES</td>
<td>Specifies that only records having a jobname which begin with the given character string(s) will be processed. This operand specifies a partial jobname from 1 to 8 characters in length. Up to 50 job groups and/or names may be specified for a single command if entered as follows: GROUPNAMES=(jobname1,...,jobnamex) The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.</td>
</tr>
<tr>
<td>JOBNAMES</td>
<td>Specifies that only records having a jobname which match the jobname(s) specified will be processed. This operand specifies a complete jobname from 1 to 8 characters in length. Up to 50 jobnames may be specified for a single command if entered as follows: JOBNAMES=(jobname1,...,jobnamex) The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.</td>
</tr>
<tr>
<td>MAXCORE</td>
<td>Specifies the maximum number of bytes of working storage available for various commands. The number may be any value from 1000 to 120000, inclusive. The default is 2400 bytes.</td>
</tr>
<tr>
<td>MAXRECLength</td>
<td>Specifies the largest SMF record that the program will process. The number may be any value from 16384 to 65536, inclusive. The default is 16384 bytes.</td>
</tr>
<tr>
<td>MAXSTACK</td>
<td>Specifies the maximum number of compressed SMF records that can be retained in storage. The number can be any value from 100 to 50000, inclusive. The default is 2500 records.</td>
</tr>
<tr>
<td>PRTLENGTH</td>
<td>Limits to the number of bytes specified to the amount of data to be printed if ERRORPRINT is indicated. The number may be any value from 32 to 65536, inclusive. The default is 32768 bytes.</td>
</tr>
<tr>
<td>Operand</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RECSIZE(rr)</td>
<td>Specifies the minimum length of the SMF records type 'rr' is to be set to the value 'nnnn'.</td>
</tr>
<tr>
<td></td>
<td>The default minimum record lengths for system generated SMF records are documented in IBM Systems Management Facilities manual.</td>
</tr>
<tr>
<td>TODATE=</td>
<td>Specifies the upper date limit of the SMF records that are to be selected, in the format yyyyddd or yyddd. If the shorter form is used, the value is prefixed with a 19.</td>
</tr>
<tr>
<td></td>
<td>The default, if the FROMDATE and/or TODATE operands are not specified, is that the date of the SMF record will not participate in the selection criteria.</td>
</tr>
</tbody>
</table>
Example A: Query

The example below provides sample JCL and control card input to find all of the activity against a group of data sets. Each SMF record encountered for a data set in the specified group will be printed. This enables you to determine all of the jobs that actively used the data set(s), based on the SMF data.

```bash
//IAMSMF  EXEC  PGM=IAMSMF,REGION=1M
//STEPLIB  DD   DISP=SHR,DSN=iam.load.lib
//SYSPRINT DD   SYSOUT=*  
//SYSMF    DD   DISP=SHR,DSN=smf.data.set
//SYSIN    DD    *
QUERY    DSG=IAMV.KSD270
/*
```

*Figure 41-14: Sample JCL for IAMSMF QUERY Command (EX4106A)*
Sample QUERY Output

```
IAM00 SMF REPORT/DATA EXTRACT PROGRAM--IAMSMF VER 9.2/00 SPIN 00 INNOVATION DATA PROCESSING DATE-2014.295 PAGE-0001
IAM303 CARD IMAGE - * QUERY DSG=IAMV.KSD270 *
IAM491 DATA SET QUERY FUNCTION STARTED - 08.41.16

DSN---IAMV.KSD270.CLUSTER  DATE--2014.295 TIME--07.18.03
JOB--------RAMTAB2 DDNAME--**ABSENT DSORG----------AM USE----------DEFINE

DSN---IAMV.KSD270.CLUSTER  DATE--2014.295 TIME--07.18.03
JOB--------RAMTAB2 DDNAME--@#$IAM DSORG----------PS USE----------OUTPUT
EXCP COUNTS---- DISK-----------1

DSN---IAMV.KSD270.CLUSTER  DATE--2014.295 TIME--07.18.04
JOB--------RAMTAB2 DDNAME--VSAMCRT1 DSORG----------PS USE----------OUTPUT
EXCP COUNTS---- DISK-----------1

DSN---IAMV.KSD270.CLUSTER  DATE--2014.295 TIME--07.18.04
JOB--------RAMTAB2 DDNAME--VSAMCRT1 DSORG----------PS USE----------OUTPUT
EXCP COUNTS---- DISK-----------2

DSN---IAMV.KSD270.CLUSTER  DATE--2014.295 TIME--07.18.04
JOB--------RAMTAB2 DDNAME--VSAMCRT1 DSORG----------PS USE----------OUTPUT
EXCP COUNTS---- DISK-----------6

DSN---IAMV.KSD270.CLUSTER  DATE--2014.295 TIME--07.18.04
JOB--------RAMTAB2 DDNAME--VSAMCRT1 DSORG----------DA USE----------OUTPUT
EXCP COUNTS---- DISK-----------14

DSN---IAMV.KSD270.CLUSTER  DATE--2014.295 TIME--07.18.04
JOB--------RAMTAB2 DDNAME--VSAMCRT1 DSORG----------DA USE----------OUTPUT
EXCP COUNTS---- DISK-----------30

DSN---IAMV.KSD270.CLUSTER  DATE--2014.295 TIME--07.18.04
JOB--------RAMTAB2 DDNAME--**ABSENT DSORG----------** USE----------SCRATCH

IAM601 SMF RECORDS -- READ.....91376 USED.....25006 DROPPED.........0
IAM492 DATA SET QUERY FUNCTION ENDED - 08.41.27 - CONDITION CODE 000
IAM499 IAMSMF( 9.2/00 SPIN 20) PROCESSING COMPLETED

Figure 41-15: Sample IAMSMF QUERY Report Output
```
IAMSMF - REPORT Command

The REPORT command is used to print job step and/or TSO session related statistics, with optional data set usage information. The reports can be used to obtain some basic performance information about selected jobs, including elapsed time, CPU time, and EXCP counts. This may be useful for comparing job performance between running with VSAM files versus running with IAM files. The reports can be requested by job or job group name from a history file (RECFM=VBS), an active SMF recording file, or from a sequential file produced by the COPY command of the program itself.

```
REPORT
    [ALLDSNAMES] [,MAXRECLENGTH= nnnn ]
    [,CHECKLENGTH ] [,MAXSTACK= nnnn]
    [,ERRORPRINT ] [,NODSNAMES ]
    [,FROMDATE= yyyyddd] [,PRTLENGTH= nnnn]
    [,FROMDDNAME= ddname ] [,RECSIZE(rrr)= nnnn ]
    [,GROUPNAMES=jobname ] [,TEMPDSNAMES]
    [,JOBNAMES= jobname ] [,TODATE= yyyyddd ]
```

Figure 41-16: IAMSMF REPORT Command Operands

**Operands**

<table>
<thead>
<tr>
<th><strong>Operand</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ALLDSNAMES</strong></td>
<td>Specifies that the IAMSMF report will show all the occurrences of any referenced data set name. The EXCP count displayed will reflect each OPEN/CLOSE. The default is that data set usage counts (EXCPs) are accumulated for each permanent data set, and the data set is only shown once. Temporary data sets are ignored. NOTE: This operand conflicts with the operand NODSNAMES.</td>
</tr>
<tr>
<td><strong>CHECKLENGTH</strong></td>
<td>Specifies that SMF data records are to be validated against a table of minimum record lengths. If the user has modified the minimum record length or is executing in a non-compatible system, the correct record lengths may be specified by the RECSIZE operand. The default is the length of an SMF record is not validated.</td>
</tr>
<tr>
<td><strong>ERRORPRINT</strong></td>
<td>Specifies that any SMF data record that causes an error during processing or fails length verification is to be printed. The default is records in error are not printed.</td>
</tr>
<tr>
<td><strong>FROMDATE=</strong></td>
<td>Specifies the lower date limit of the SMF records that are to be selected, in the form yyyyddd or yyyddd. The shorter form is prefixed with a 19. The default if the FROMDATE and/or TODATE operands are not specified, is that the date of the SMF record will not participate in the selection criteria.</td>
</tr>
<tr>
<td><strong>Operand</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>FROMDDNAME</strong></td>
<td>Specifies the DDNAME of the SMF file to be used as input to IAMSMF. The default input DDNAME is SYSMF.</td>
</tr>
<tr>
<td><strong>GROUPNAMES</strong></td>
<td>Specifies that only those records having a jobname which begin with the specified character string(s) will be processed. This operand specifies a partial jobname from 1 to 8 characters in length. Up to 50 job groups and/or names may be specified for a single command if entered as follows: GROUPNAMES=(jobname1,...,jobnamex) The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.</td>
</tr>
<tr>
<td><strong>JOBNAMES</strong></td>
<td>Specifies that only those records having a jobname which match the jobname(s) specified will be processed. This operand specifies a complete jobname from 1 to 8 characters in length. Up to 50 job names may be specified for a single command if entered as follows: JOBNAMES=(jobname1,...,jobnamex) The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.</td>
</tr>
<tr>
<td><strong>MAXRECLength</strong></td>
<td>Specifies the largest SMF record that the program will process. The number may be any value from 16384 to 65536, inclusive. The default is 16384 bytes.</td>
</tr>
<tr>
<td><strong>MAXSTACK</strong></td>
<td>Specifies the maximum number of compressed SMF records that can be retained in storage. The number can be any value from 100 to 50000, inclusive. The default is 2500 records.</td>
</tr>
<tr>
<td><strong>NODSNAMES</strong></td>
<td>Specifies that the report will not show data set oriented information.</td>
</tr>
<tr>
<td><strong>PRTLLENGTH</strong></td>
<td>Limits to the specified amount of data to be printed if ERRORPRINT is indicated. The number may be any value from 32 to 65536, inclusive. The default is 32768 bytes.</td>
</tr>
<tr>
<td><strong>RECSIZE(rrr)</strong></td>
<td>Specifies the minimum length of the specified SMF record type is to be set to the value specified. The default minimum record lengths for system generated SMF records are documented in the IBM Systems Management Facilities manual.</td>
</tr>
</tbody>
</table>

**NOTE:** This operand conflicts with the operands ALLDSNAMES and TEMPDSNAMES.
<table>
<thead>
<tr>
<th><strong>Operand</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TEMPDSNAMES</strong></td>
<td>Specifies that reports produced will contain information on temporary as well as permanent data sets. By default, temporary data sets are ignored. <strong>NOTE:</strong> This operand conflicts with the operand NODSNAMES.</td>
</tr>
<tr>
<td><strong>TODATE=</strong></td>
<td>Specifies the upper date limit of the SMF records that are to be selected, in the form yyyyddd or yyddd. The shorter form is prefixed with a 19. The default, is that there is no upper date limit on the SMF records to be processed.</td>
</tr>
</tbody>
</table>
Example A: IAMSMF REPORT Command

The following JCL example demonstrates how to run the IAMSMF REPORT command. For this request, the GROUP keyword is specified with a value of KSD. This will cause IAMSMF to report on all jobs that have a job name beginning with KSD that are contained within the SMF data provided.

```
//IAMSMF EXEC PGM=IAMSMF,REGION=1M
//STEPLIB DD DISP=SHR,DSN=iam.load.lib
//SYSPRINT DD SYSOUT=* 
//SYSMF DD DISP=SHR,DSN=smf.data.set
//SYSIN DD *
REPORT GROUP=KSD
/*
```

Figure 41-17: Example of JCL to run IAMSMF REPORT Command (EX4107A)

Sample IAMSMF REPORT Output

Shown below is a sample of what the REPORT output looks like, having been run with the above sample control card and JCL. Explanations of the data fields that appear on this report are described after the report example.

```
IAM400 SMF REPORT/DATA EXTRACT PROGRAM -- IAMSMF VER 9.2/00 SPIN 00 INNOVATION DATA PROCESSING DATE-2014.295
IAM303 CARD IMAGE - * REPORT GROUP=KSD
IAM491 SMF REPORT FUNCTION STARTED - 08.57.33

JOB NAME........KSD270 DATE....2014.295
STEP NAME...KSD270A PGM....IDCAMS REGION REQ...3072 REGION USE..14508 COMP CODE..C-0000 PERFORM......0
START..08.57.13 STOP...08.57.14 WALL..00.00.00.29 CPU...00.00.00.09 SRB...00.00.00.00
TAT...00.00.00.25 TNA...00.00.00.04 TRT...00.00.00.25
DD..@#$IAM JOBLIB IAMX.IAM92.TESTLIB
DSNAME.....IAMV.KSD270.CLUSTER PS USE......OUTPUT DISK........1
SUMMARY OF I/O ACTIVITY....... PAGE IN.........0 PAGE OUT........0 TAPE.........0 DISK.......99
CONN.....454144 SWAP IN.........0 SWAP OUT........0 SWAP COUNT......0 SRVU........633
STEP NAME...KSD270B PGM....IAMTVSAM REGION REQ...3072 REGION USE..15868 COMP CODE..C-0000 PERFORM......0
START..08.57.14 STOP...08.57.15 WALL..00.00.01.07 CPU...00.00.00.17 SRB...00.00.00.00
TAT...00.00.00.89 TNA...00.00.00.18 TRT...00.00.00.89
DD..VSAMCRT1 JOBLIB IAMX.IAM92.TESTLIB
DSNAME.....IAMV.KSD270.CLUSTER DA USE......OUTPUT DISK........30
SUMMARY OF I/O ACTIVITY....... PAGE IN.........0 PAGE OUT........0 TAPE.........0 DISK.......384
CONN.....162048 SWAP IN.........0 SWAP OUT........0 SWAP COUNT......0 SRVU.......1259
```

Figure 41-18: Sample of an IAMSMF REPORT Command Output
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Name</td>
<td>Provides the name of the JOB being reported on.</td>
</tr>
<tr>
<td>Date</td>
<td>The date on which the job ran.</td>
</tr>
<tr>
<td>Step Name</td>
<td>For each job step, this specifies the name of that step.</td>
</tr>
<tr>
<td>PGM</td>
<td>Indicates the name of the program being executed, from the EXEC PGM= card.</td>
</tr>
<tr>
<td>Region REQ</td>
<td>Shows the amount of region below the line requested by this job step.</td>
</tr>
<tr>
<td>Region Use</td>
<td>Shows the actual amount of virtual storage used by this job step.</td>
</tr>
<tr>
<td>Comp Code</td>
<td>Indicates the completion code for the job step.</td>
</tr>
<tr>
<td>Perform</td>
<td>Indicates the performance group that the job step ran in.</td>
</tr>
<tr>
<td>Start</td>
<td>Starting time of the job step.</td>
</tr>
<tr>
<td>Stop</td>
<td>Ending time of the job step.</td>
</tr>
<tr>
<td>Wall</td>
<td>Elapsed time of the job step.</td>
</tr>
<tr>
<td>CPU</td>
<td>Total TCB time used by the job step.</td>
</tr>
<tr>
<td>SRB</td>
<td>Total SRB time used by the job step.</td>
</tr>
<tr>
<td>TAT</td>
<td>Transaction Active Time. This is defined as the time that the transaction was swapped in plus the time that the transaction was swapped out but ready to run.</td>
</tr>
<tr>
<td>TNA</td>
<td>Transaction Not Active Time, which is calculated by subtracting the Transaction Active Time from the total elapsed time of the job step.</td>
</tr>
<tr>
<td>TRT</td>
<td>The Transaction Residence Time from the SMF record. This is defined as being the time that the transaction was swapped in.</td>
</tr>
<tr>
<td>DD</td>
<td>Specifies the DD NAME detail. A line is included for each permanent DISK or TAPE data set.</td>
</tr>
<tr>
<td>DSNAME</td>
<td>The name of the data set referenced on the indicated DD card.</td>
</tr>
<tr>
<td>DSORG</td>
<td>The data set organization.</td>
</tr>
<tr>
<td>USE</td>
<td>How the data set was used, e.g. as INPUT or OUTPUT.</td>
</tr>
<tr>
<td>DISK or TAPE</td>
<td>Indicates the type of device that the data set was on followed by the EXCP count.</td>
</tr>
<tr>
<td>Page In</td>
<td>The number of pages of virtual storage that were read from the paging data set.</td>
</tr>
<tr>
<td>Page Out</td>
<td>The number of pages of virtual storage that were written to a paging data set.</td>
</tr>
<tr>
<td>Tape</td>
<td>The total number of TAPE EXCP's.</td>
</tr>
<tr>
<td>Disk</td>
<td>The total number of DISK EXCP's.</td>
</tr>
<tr>
<td>CONN</td>
<td>Connect Time</td>
</tr>
<tr>
<td>Swap In</td>
<td>The total number of pages swapped in.</td>
</tr>
<tr>
<td>Swap Out</td>
<td>The total number of pages swapped out.</td>
</tr>
<tr>
<td>Swap Count</td>
<td>The number of times that the job step was swapped out.</td>
</tr>
<tr>
<td>SRVU</td>
<td>The number of Service Units used by the job step.</td>
</tr>
</tbody>
</table>
45 IAMRECVR – IAM DATASET RECOVERY

45.01 IAMRECVR – Recovery Program Overview

Overview

To assist in recovery of valuable data, IAM includes a special function program, IAMRECVR, which can often create a backup copy of the data in an IAM file that is unreadable by normal means. The IAMRECVR program has been designed to aid users in the recovery of the contents of IAM files that have become unusable. Such problems can arise due to hardware or media failures, due to improper sharing of the data set for update, or due to an application or system software error or failure.

IAMRECVR is a tool to assist in the recovery of IAM data sets. IAMRECVR is not a replacement for having established data recovery procedures, but rather is a program that can be used as a part of data recovery. It is quite important to have procedures in place to recover data should unforeseen problems occur, which include periodic backups of the data set, and possibly transaction journals or logs of updates to the data set.

IAMRECVR reads the IAM file using a high performance EXCP technique. Only blocks determined to match the IAM file specifications are processed. Any block determined to be unreadable due to physical damage to the device (e.g.: data check) or due to corruption of the record formatting within the block will be skipped. Appropriate error messages describing any of the errors encountered will be printed.

On a recovery process, if there are records in Independent or Extended Overflow, the sequential output file will need to be sorted. IAMRECVR can do that automatically, and will then report on any duplicate records encountered, which can optionally be written out to a log data set. The sequential output data set from IAMRECVR can subsequently be used to REPRO the data back into the recovered data set.

Other capabilities offered by IAMRECVR include:

- Decompress and write out a file of uncompressed data records from a backup of an IAM file that was done with the BACKUPCOMPRESSED feature.
- Validate the integrity of the data blocks within an IAM data set.
- Print out portions of the IAM data set in a dump format.
- Create an image of the file structure of an IAM data set without copying any data other than the key.
- Produce a report similar to a LISTCAT with the IAM file characteristics and statistics.

*If you are preparing to do an IAM file recovery, be sure to read “10.87 Recovering IAM Datasets” on page 195. That section provides an explanation, a technique, and several examples of using IAMRECVR to recover an IAM data set.*
The IAMRECVR program has the following commands / functions.

- **APPLY**: Copy records from a SPANOUT or LOG data set created by the RECOVER command, into an IAM or VSAM data set.
- **DIAGNOSE**: Validates the basic data integrity of an IAM file.
- **IAMSTRUCTURE**: Dump the structure of an IAM data set. The only portion of the data retained in the backup will be the key.
- **LIST**: Display the attributes of an IAM data set.
- **PRINT**: Print out selected portions or an entire IAM data set, in a dump format.
- **RECOVER**: Reads a damaged IAM file, producing a sequential (or IAM / VSAM) data set containing the records that IAMRECVR is able to read.
The JCL statements required to execute IAMRECVR are as follows:

**Execute Statement**

Specifies the name of the IAM recovery program -- IAMRECVR. For a file recovery operation, sufficient storage on the REGION parameter must be specified to include storage for a SORT, which may need to be called if there are records in Extended or Independent Overflow. A typical execute statement would be:

```bash
//RECOVER EXEC PGM=IAMRECVR,REGION=2048K
```

**DD Statements**

The following table identifies the DD statements required for running IAMRECVR.

<table>
<thead>
<tr>
<th>DD Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB or JOBLIB</td>
<td>An optional DD statement that specifies the library containing the IAM program load modules. This DD is not necessary if IAM is in the Link List, as is recommended. If a STEPLIB or JOBLIB is used, then the IAM load library must be APF authorized.</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>Specifies where the IAMRECVR messages are to be printed. Usually a SYSOUT data set.</td>
</tr>
<tr>
<td>DISKIN</td>
<td>Specifies the IAM data set to be recovered, listed or printed.</td>
</tr>
<tr>
<td>DISKOUT</td>
<td>Specifies the new IAM file to be created from the damaged IAM file. The new IAM file will have the same characteristics as the IAM file being recovered.</td>
</tr>
<tr>
<td>TAPEOUT</td>
<td>Specifies the sequential recovery file to be created by IAMRECVR on either a tape or direct access device. If a direct access device is used, enough space must be allocated to contain a sequential copy of the IAM file. The use of secondary allocation values is permitted and encouraged.</td>
</tr>
<tr>
<td>VSAMOUT</td>
<td>Specifies the new VSAM file to be loaded from the damaged IAM file.</td>
</tr>
<tr>
<td>SPANOUT</td>
<td>Specifies a sequential data set used to hold spanned records for recovery. This file can reside on either tape or direct access, and will be used by the APPLY command after reloading the recovered data set containing the non-spanned records from the TAPEOUT file.</td>
</tr>
<tr>
<td>LOG</td>
<td>For the RECOVER option, specifies a sequential file where the duplicates, if any, are to be stored. For the APPLY option, specifies the sequential file to be used as input.</td>
</tr>
<tr>
<td>SYSIN</td>
<td>Specifies the input control statement data set. Usually a DD * data set.</td>
</tr>
<tr>
<td>SORTWKnn</td>
<td>These DD statements may be required if you are running the RECOVER command. Be sure to provide adequate SORT work space for the file that you are recovering. Refer to documentation of your sort for additional information.</td>
</tr>
<tr>
<td>SORTLIB</td>
<td></td>
</tr>
</tbody>
</table>
IAMRECVR - APPLY Command

The APPLY command reads the LOG data set created by the RECOVER command, and will either add or replace records in the IAM data set from the LOG data set. Apply is similar in function to an IDCAMS REPRO REPLACE. The APPLY command is used when a RECOVER command found duplicate records in the original IAM data set to copy those records into the recovered data set. The duplicate records will be copied from the LOG data set that was created by the RECOVER command if DUPLICATES=LOG was specified.

For more information on how the APPLY command fits into the recovery process for an IAM data set, refer to Section 10.87 Recovering IAM Data Sets.

```
APPLY
  [AUDIT= ccccc] [,IAMDDNAME= ddname ]
  [,LOGDDNAME=ddname] [,OUTPUTFILE= cccc ]
  [,PRTLENGTH= nnnnn ] [,RRDS]
  [SPANDDNAME= ddname ] [,SPANNED ]
  [,VSAMDDNAME= ddname ]
```

Figure 45-1: IAMRECVR APPLY Command Operands
### IAMRECVR - APPLY Command

#### Operands

<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AUDIT=</strong></td>
<td>Defines the audit trail requirements for records processed by the APPLY command. Possible values are:</td>
</tr>
<tr>
<td></td>
<td><strong>KEY</strong> - Print the key from the data record.</td>
</tr>
<tr>
<td></td>
<td><strong>NONE</strong> - No audit trail is produced.</td>
</tr>
<tr>
<td></td>
<td><strong>RECORD</strong> - Print the entire data record.</td>
</tr>
<tr>
<td></td>
<td>The default is KEY.</td>
</tr>
<tr>
<td><strong>IAMDDNAME=</strong></td>
<td>Defines the DDNAME of the IAM file that is to be updated.</td>
</tr>
<tr>
<td></td>
<td>The default is DISKOUT.</td>
</tr>
<tr>
<td><strong>LOGDDNAME=</strong></td>
<td>Defines the DD Name of the log file to be used as input to the APPLY operation. This file is from the LOG output of a RECOVER.</td>
</tr>
<tr>
<td></td>
<td>The default DD Name is LOG.</td>
</tr>
<tr>
<td><strong>OUTPUTFILE=</strong></td>
<td>Specifies the access method to use for the target data set. Valid values are:</td>
</tr>
<tr>
<td></td>
<td><strong>IAM</strong> – Use the native mode IAM interface. Can not be used for Enhanced format files, must specify VSAM.</td>
</tr>
<tr>
<td></td>
<td><strong>VSAM</strong> – This can be specified for either IAM or VSAM data sets. This option will cause IAMRECVR to use VSAM I/O macros.</td>
</tr>
<tr>
<td><strong>PRTLENGTH=</strong></td>
<td>Limit the amount of information printed in the AUDIT trail to this value or the length of the key or data, whichever is smaller.</td>
</tr>
<tr>
<td></td>
<td>The default is 32768.</td>
</tr>
<tr>
<td><strong>RRDS</strong></td>
<td>A keyword that indicates an RRDS is the target of the APPLY and that the input dataset contains the Relative Record Number for each record. This corresponds to the use of the KEEPRRN keyword on the RECOVER command.</td>
</tr>
<tr>
<td><strong>SPANDDNAME=</strong></td>
<td>Specifies the DD name of the data set containing the spanned records. This is the data set that was created by the IAMRECVR RECOVER command.</td>
</tr>
<tr>
<td></td>
<td>The default is SPANOUT.</td>
</tr>
<tr>
<td><strong>SPANNE</strong></td>
<td>Specifies that the APPLY command is to apply the spanned records contained in the data set specified by the SPANOUT DD card. The LOG dataset will not be used in the APPLY SPANNED operation.</td>
</tr>
<tr>
<td><strong>VSAMDDNAME=</strong></td>
<td>Specifies the DD name of the IAM or VSAM data set that is to be updated when OUTPUTFILE=VSAM is specified.</td>
</tr>
<tr>
<td></td>
<td>The default is VSAMOUT.</td>
</tr>
</tbody>
</table>
Example A: Apply Command

A basic example of using the APPLY command is shown below. For information on how the APPLY command should be used for a file recovery, refer to Section 10.87, Recovering an IAM Data Set.

```
//APPLYLOG EXEC PGM=IAMRECVR,REGION=4M
//SYSPRINT DD   SYSOUT=*  
//DISKOUT DD   DISP=OLD,DSN=my.iam.new.dataset  
//LOG      DD   DISP=SHR,DSN=my.iam.log.dataset  
//SYSIN    DD   *  
APPLY  
/*
```

Figure 45-2: Example of JCL to run the APPLY Command (EX4503A)

Example B: Apply Spanned Records

Shown below is an example of using the APPLY command to add the spanned records back into the recovered data set. Prior to executing this APPLY command, the file must be reloaded with the non-spanned records from the TAPEOUT file. Subsequently, the following APPLY command can be used to put the spanned records back into the recovered data set.

```
//APPLYSPN EXEC PGM=IAMRECVR,REGION=4M  
//SYSPRINT DD   SYSOUT=*  
//DISKOUT DD   DISP=OLD,DSN=my.iam.new.dataset  
//SPANOUT DD   DISP=SHR,DSN=my.iam.spanout.dataset  
//SYSIN    DD   *  
APPLYSPANNED  
/*
```

Figure 45-3: Example of JCL to run the APPLY SPANNED Command (EX4503B)
IAMRECVR – DECOMPRESS Command

Recovery of Data that is Compressed

The IAM file recovery program, IAMRECVR is able to read sequential files created with the BACKUPCOMPRESSED feature, and write out a sequential file with the data uncompressed. This may be useful for when an application program needs to read the sequential file from FDRREORG or IDCAMS REPRO of the IAM file. This is facilitated with the new command, DECOMPRESS.

DECOMPRESS Command Statement

The DECOMPRESS command allows a compressed backup of an IAM file to be decompressed in the event that IAM is not available. The IAM VSAM Interface (VIF) does not have to be active.

```
DECOMPRESS
    [ FROMDDNAME= ddname ] ,KEYLEN= nnn
    ,RKP= nnnnn [,SCAN]
    [,TODDNAME= ddname ]
```

*Figure 45-4: IAMRECVR DECOMPRESS Command Operands*

DECOMPRESS Command Operands

The table below contains descriptions of the keywords for the DECOMPRESS command of IAMRECVR. The minimal abbreviation for each operand keyword is underlined.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FROMDDNAME=</td>
<td>Defines the DDNAME of the compressed IAM backup file. The default is DISKIN.</td>
</tr>
<tr>
<td>KEYLEN=</td>
<td>Required operand that specifies the length of the key within the data records in the IAM file. This value can be obtained from an IAMINFO or LISTCAT report on the original IAM data set.</td>
</tr>
<tr>
<td>RKP=</td>
<td>Required operand that specifies the relative location of the key within the data record in the IAM file. This value can be obtained from an IAMIFNO or LISTCAT report on the original IAM data set.</td>
</tr>
<tr>
<td>SCAN</td>
<td>Allows a compressed backup file to be read and all records decompressed, but an output file will not be created. SCAN can be used to verify a compressed backup file.</td>
</tr>
<tr>
<td>TODDNAME=</td>
<td>Defines the DDNAME of the output uncompressed flat file. The default is TAPEOUT.</td>
</tr>
</tbody>
</table>
EXAMPLE A: DECOMPRESS

Assume an IAM file with 100 byte records, an 8 byte key length with a relative key position (RKP) of 10 has been backed up by an IDCAMS REPRO with the BACKUPCOMPRESSED option, and is now needed in its uncompressed state.

```
//DECOMPRSEXEC=IAMRECVR
//SYSPRINTDDSYSOUT=*  
//DISKINDDSDSN=my.seqfile,DISP=OLD
//TAPEOUTDDDSN=my.uncomp.seqfile,DISP=(,CATLG),
//DCB=(RECFM=VB,LRECL=104,BLKSIZ=32760),UNIT=TAPE
//SYSINDD*
    DECOMPRESSKEYLEN=8,RKP=10
/*
```

Figure 45-5: Example of JCL and Control Card for Decompress (EX4504A)
IAMRECVR - DIAGNOSE Command

The DIAGNOSE command will read through the entire IAM file, and report on any errors that are encountered. The DIAGNOSE function reads the IAM data set without using the IAM access method, and verifies the read integrity of each data block. DIAGNOSE can detect physical I/O errors, validate the format of all blocks containing user data, and verify that all records can be uncompressed.

The DIAGNOSE command is not able to detect problems with the control information saved for Independent Overflow, or the Extended Index. The DIAGNOSE will detect out of sequence records and duplicate records contained within any particular data block, however it is not able to detect for duplicate records that may exist in other areas of the file, such as in the Overflow areas.

```
DIAGNOSE [FROMDDNAME= ddname ]
```

Figure 45-6: IAMRECVR DIAGNOSE Command Operands

The following operand may be specified with the DIAGNOSE subcommand.

**Operand** | **Description**
--- | ---
FROMDDNAME= | Defines the DDNAME of the IAM file that is to be diagnosed.

The default is DISKIN.

**EXAMPLE A:** The following example demonstrates how to run an IAMRECVR DIAGNOSE command.

```
//DIAGNOSE EXEC PGM=IAMRECVR
//SYSPRINT DD SYSOUT=*
//DISKIN DD DISP=SHR,DSN=my.iam.data.set
//SYSIN DD *
DIAGNOSE
/*
```

Figure 45-7: Example of the IAMRECVR Diagnose Command (EX4505A)
The IAMSTRUCTURE command is used to copy all IAM KEY and INDEX information to a sequential output file. This output file can then be sent to Innovation to recreate as closely as possible the users file, without having to send any actual data. IAMSTRUCTURE will copy the entire KEY of each record, and the record length.

IAMSTRUCTURE will allow Innovation to recreate the logical structure of the IAM file to help with problem determination, and for testing of new enhancements to IAM. The output of the IAMSTRUCTURE command is a physical sequential (PS) file, which can be browsed under TSO to verify that no confidential information is being copied.

The IAMSTRUCTURE command has no operands. The input IAM file is specified on the DISKIN DD statement, and the sequential output file is specified on the TAPEOUT DD statement. An example of the JCL and control card to run IAMSTRUCTURE is shown below.

```
//DUMPSTRC EXEC PGM=IAMRECVR,REGION=4M
//SYSPRINT DD   SYSOUT=* 
//DISKIN   DD   DISP=SHR,DSN=my.iam.file
//TAPEOUT  DD   DISP=(,CATLG),DSN=my.iam.structre.file,
//              UNIT=SYSDA,SPACE=(CYL,(5,2))
//SYSIN    DD   *
IAMSTRUCTURE
/*
```

*Figure 45-8: Sample JCL for the IAMSTRUCTURE Command (EX4506A)*
LIST Command

45.07 LIST Command

The LIST command is used to list the characteristics of an IAM file. The format of the report is almost identical to the report on IAMPRINT from a LISTCAT ALL. The LISTCAT ALL will provide more information than the IAMRECVR LIST command, so use of a LISTCAT is recommended above the IAMRECVR LIST command.

```
LIST [FROMDDNAME=  ddname ]
```

Figure 45-9: IAMRECVR LIST Command Operands

LIST Command Operands

The following operand may be specified with the LIST command.

Operand Description
FROMDDNAME= Defines the DDNAME of the IAM file that is to be listed.
The default is DISKIN.

```
//LISTIAMF EXEC PGM=IAMRECVR,REGION=1M
//SYSPRINT DD   SYSOUT=*
//DISKIN   DD   DISP=SHR,DSN=my.iam.file
//SYSIN    DD   *
LIST
/*
```

Figure 45-10: Example of JCL for LIST Command (EX4507A)
45.08 IAMRECVR - PRINT Command

The PRINT command is used to print various areas or blocks in dump format from an IAM file. A list of the file characteristics is produced upon completion.

```
PRINT
    [ALLBLKS]   [,DATA ]
    [,EXKEYS]   [,EXTENDED]
    [,FBLK= nnnnn]   [,FROMDDNAME= ddname]
    [,IDPINQ ]   [,KEYS ]
    [,MAXBLKS= nnnnn]   [,OFLOW ]
    [,PE ]       [,PRTLENGTH= nnnnn ]
    [,TBLK= nnnnn ]
```

Figure 45-11: IAMRECVR PRINT Command Operands

**Operands**

**Operand** | **Description**
--- | ---
**ALLBLKS** | Specifies that all (or portions of all) blocks in the IAM file are to be printed.

NOTE: Use of this operand conflicts with DATA, FBLK, KEYS, OFLOW, PE, and TBLK.

**DATA** | Specifies that all (or portions of all) prime data blocks in the IAM file are to be printed.

NOTE: Use of this operand conflicts with ALLBLKS, FBLK, and TBLK.

**EXKEYS** | For Enhanced Format files, specifies that the PE index blocks and the Overflow RBN blocks are to be printed.

**EXTENDED** | For Enhanced Format files, specifies that all of the blocks (data and index) in the extended area of the file will be printed. These include Extended Overflow and PE blocks.

**FBLK=** | Specifies the From block number, relative to 1, from which printing is to begin.

NOTE: Use of this operand conflicts with ALLBLKS, DATA, KEYS, OFLOW, and PE.

**FROMDDNAME=** | Defines the DDNAME of the IAM file that is to be listed.

The default is DISKIN.

**IDPINQ** | Specifies the IAM control block is to be printed.

**KEYS** | Specifies that all (or portions of all) key blocks in the IAM file are to be printed.

NOTE: Use of this operand conflicts with ALLBLKS, FBLK, and TBLK.
**MAXBLKS=** Specifies the maximum number of blocks to be printed from each area selected in the IAM file.

The default is the number of blocks in the area selected or the entire file if ALLBLKS is specified, excluding the IAM control block.

**OFLOW** Specifies that all (or portions of all) Independent Overflow blocks in the IAM file are to be printed. NOTE: Use of this operand conflicts with 'ALLBLKS', 'FBLK', and 'TBLK'.

**PE** Specifies that all (or portions of all) Prime Extension blocks in the IAM file are to be printed. NOTE: Use of this operand conflicts with 'ALLBLKS', 'FBLK', and 'TBLK'.

**PRTLENGTH=** Limit the amount of data printed for each block to this value or the length of block, whichever is smaller. The default is 32768.

**TBLK=** Specifies the block number, relative to 1, at which printing is to end. NOTE: Use of this operand conflicts with 'ALLBLKS', 'DATA', 'KEYS', 'MAXBLKS', 'OFLOW', and 'PE'.

**EXAMPLE A:**
The example below demonstrates two different forms of the PRINT command. The first prints out the IAM control information blocks, in a dump format. The second prints out selected data blocks from the IAM data set. The use of the PRINT command in problem diagnosis and recovery is discussed in Section 10.87, Recovering an IAM Data Set.

```
//PRINTIAM EXEC PGM=IAMRECVR,REGION=4M
//SYSPRINT DD   SYSOUT=*  
//DISKIN   DD   DISP=SHR,DSN=my.iam.dataset
//SYSIN    DD   *
PRINT IDPINQ
PRINTFBLK=100,MAXBLKS=10
/*
```

*Figure 45-12: Example of the Print Command under IAMRECVR (EX4508A)*
45.09 IAMRECVR - RECOVER Command

The RECOVER command is used to read an IAM data set, which may be damaged or corrupted, and copy the records it is able to read into another data set. Any errors encountered reading the input IAM data set will be reported on, and may result in the loss of some data records if the errors cause some of the records to be unreadable. The output data set can be a sequential data set, an IAM data set, a VSAM cluster, or a combination of sequential and IAM or VSAM. While there are several choices for the type of output data set, Innovation recommends using only a sequential output data set. The sequential data set can then be copied into an IAM or VSAM data set using IDCAMS REPRO.

The RECOVER command may not detect some of the errors, particularly if they occur within the overflow control information or extended index areas. Such errors may result in being unable to open the IAM data set through normal programs. IAMRECVR does not rely on that information to open or read the data set. The RECOVER will still work and be valid, even though no errors were detected within the data blocks.

If the input IAM data set has records in Extended Overflow, or for Compatible format files in Independent Overflow, the output file will have to be sorted. The RECOVER command can automatically invoke the SORT, and it is highly recommended that the SORT be done by IAMRECVR. You will need to provide IAMRECVR with sufficient SORT work space based on the size of the file that is being recovered.

For some good examples of procedures to follow when recovering IAM data sets, be sure to read Section 10.87 of the manual, Recovering IAM Data Sets. A full explanation of how to use the RECOVER command is provided there, along with several examples.

```
RECOVER
[AUDIT= cccccc ] [,BLKSIZE= nnnnn ]
[,COMPRESSED] [,DUPLICATES= cccccc ]
[,FROMDDNAME= ddname ] [,IAMDDNAME= ddname ]
[,KEEPRRN ] [,KEYLEN= nnn ]
[,LOGDDNAME= ddname ] [,LRECL= nnnnn ]
[,MAXBLKS= n...n ] [,NOPRIMESORT]
[,OUTPUTFILES= cccc ] [,OVERFLOW= nnnnn ]
[,PRTLENGTH=nnnnn ] [,RKP= nnnn ]
[,SORT= cccccc ] [,SORTCORE= n...n ]
[,SORTMSG= cc ] [,SORTPFX= cccc ]
[,SPANDDNAME= ddname ] [,TODDNAME= ddname ]
[,VARIABLE ] [,VSAMDDNAME= ddname ]
```

Figure 45-13: IAMRECVR Recover Command Operands
## IAMRECVR - RECOVER Command

### Operands

<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AUDIT=</strong></td>
<td>Defines the audit trail requirements for duplicate records processed by the RECOVER command if SORT=IFREQ or SORT=YES and DUPLICATE= APPLY or if DUPLICATE= PRINT or DUPLICATE=LOG are specified.</td>
</tr>
<tr>
<td><strong>BLKSIZE=</strong></td>
<td>Specifies the actual blocksize of the IAM file. This value can be obtained from the runtime statistics or a LIST command.</td>
</tr>
<tr>
<td><strong>COMPRESSED</strong></td>
<td>Identifies the file as containing compressed records.</td>
</tr>
<tr>
<td><strong>DUPLECTES=</strong></td>
<td>Defines the processing requirements for any duplicate records processed by the RECOVER command if SORT=IFREQ or SORT=YES is specified.</td>
</tr>
<tr>
<td><strong>FROMDDNAME=</strong></td>
<td>Defines the DDNAME of the IAM file that is to be recovered.</td>
</tr>
<tr>
<td><strong>IAMDDNAME=</strong></td>
<td>Defines the DDNAME of the IAM file that is to be created when OUTPUTFILES=IAM or BOTH is specified.</td>
</tr>
</tbody>
</table>

**AUDIT=**

- **KEY** - Print the key from the data record.
- **NONE** - No audit trail is produced.
- **RECORD** - Print the entire data record.

The default is **KEY**.

**BLKSIZE=**

Specifies the actual blocksize of the IAM file. This value can be obtained from the runtime statistics or a LIST command.

**NOTE:** This value is ignored unless the IAM control record is destroyed, at which time it is required.

**COMPRESSED**

Identifies the file as containing compressed records.

**NOTE:** This value is ignored unless the IAM control record is destroyed. This value is optional and is only used to request that the new IAM file is to have a compressed data structure. IAMRECVR can always detect a compressed record and decompress it.

**DUPLECTES=**

Defines the processing requirements for any duplicate records processed by the RECOVER command if SORT=IFREQ or SORT=YES is specified.

- **APPLY** - Update the IAM file being created with the duplicate records. Ignored unless 'OUTPUTFILES=IAM' or 'OUTPUTFILES=BOTH' is specified.
- **IGNORE** - Ignore duplicate records.
- **LOG** - Create a log data set of any duplicate records. This data set may later be used as input the 'APPLY' command.
- **PRINT** - Print any duplicate records.

The default is **PRINT**.

**FROMDDNAME=**

Defines the DDNAME of the IAM file that is to be recovered.

The default is **DISKIN**.

**IAMDDNAME=**

Defines the DDNAME of the IAM file that is to be created when OUTPUTFILES=IAM or BOTH is specified.

The default is **DISKOUT**.

**NOTE:** For Compatible format files, IAMRECVR will use the native IAM interface. For Enhanced format files, IAMRECVR uses the IAM VIF interface, so the file will have to be defined prior to running the RECOVER.
**Operand** | **Description**
--- | ---
**KEEPRRN** | For RRDS datasets, specifies that the Relative Record Number (RRN) will be included with the data for each record that is written to the output file. When specified on the RECOVER control card include the RRDS keyword on the subsequent APPLY command for the record number to be used when recovering the dataset.

**KEYLEN=** | Specifies the length of the key within the data records in the IAM file. This value can be obtained from the run time statistics or a LIST command.

**NOTE:** This value is ignored unless the IAM control record is destroyed, at which time it is required.

**LOGDDNAME=** | Defines the DDNAME of the log file to be created.

The default is LOG.

**LRECL=** | Specifies the logical record length of the data records in the IAM file. This value can be obtained from the run time statistics or a LIST command.

**NOTE:** This value is ignored unless the IAM control record is destroyed, at which time it is required.

**MAXBLKS=** | Specifies the number of data blocks in the IAM file. This value can be obtained from the run time statistics or a 'LIST' command.

**NOTE:** This value is ignored unless the IAM control record is destroyed, at which time it is required.

**NOrmPRIME SORT** | Useful when large IAM files contains records in overflow, which normally requires a sort of all records. If the IAM file is very large, it may be difficult to obtain enough SORTWORK space for this. Since all records in the IAM Prime Data Area are in sorted order, the only records that need to be sorted are in Extended Overflow. This option causes two sequential output files to be created – one containing the records from the Prime Data Area and the other containing the sorted records from Extended Overflow. Once the records from the Prime Data Area have been reloaded into the new IAM file, the sorted records from Extended Overflow can be added back via REPRO REUSE or by using the APPLY command of IAMRECVR.

**OUTPUTFILES=** | Defines the output requirements for the RECOVER subcommand. Valid values are:

**BOTH** - Create both an IAM file and a sequential copy.

**BOTHV** - Create both an IAM file (using the VSAM interface) or a VSAM file, and a sequential copy.

**IAM** - Create only an IAM file.

**SEQ** - Create only a sequential copy of the recoverable data remaining in the IAM file.

**VSAM** - Create an IAM file (using the VSAM interface) or a VSAM file.

The default is SEQ.
OVERFLOW= Specifies the number of Independent Overflow blocks in the IAM file. This value can be obtained from the run time statistics or a LIST command.

NOTE: This value is ignored unless the IAM control record is destroyed, at which time it is required.

Operand Description

PRTLENGTH= Limit the amount of data printed for each block to this value or the length of block, which ever is smaller.

The default is 32768.
Specifies the relative location of the key within a data record in the IAM file. This value can be obtained from the run time statistics or a LIST command.

RKP= NOTE: This value is ignored unless the IAM control record is destroyed, at which time it is required.
Defines the output sorting requirements for the RECOVER command. Valid values are:

IFREQ - Sort the records only if sequence checks are encountered in the file.

SORT= NO - Do not sort the records.

YES - Sort the records.
The default is NO.

SORTCORE= Specifies the amount of storage the program SORT is to use if external sorting is required. The number maybe from 10000 to 8000000 inclusive.
The default is 100000.
Specifies the message option to be used by the program SORT if external sorting is required.

AC - All messages to the console

AP - All messages to the printer (SYSOUT)

CC - Critical messages to the console

CP - Critical messages to the printer

NO - No messages to be produced

PC - Critical messages to both console and printer
The default is CC.

SORTMSG= Specifies the DDNAME prefix to be used by the program SORT if external sorting is required. If the string specified is less than 4 characters, a dollar sign ($) fill character will be used.
The default is SORT.
**SPANDDNAME**

Defines the DDNAME of the sequential output data set created during recovery that contains the spanned records.

The default is SPANOUT.

**TOADDNAME**

Defines the DDNAME of the sequential output data set created during recovery.

The default is TAPEOUT.

**VARIABLE**

Identifies the IAM file as having variable length records.

NOTE: This value is ignored unless the IAM control record is destroyed, at which time it is required.

**VSAMDDNAME**

Defines the DDNAME of the IAM or VSAM file to be created when OUTPUTFILES=VSAM or BOTHV is specified.

The default is VSAMOUT.
EXAMPLE A: Recover

The example below shows a basic RECOVER operation to a sequential data set. Subsequent to the RECOVER, the sequential data would be copied into an IAM data set using an IDCAMS REPRO. While it is rare that there will be duplicate records, they do occur on occasion. To save time when such a circumstance occurs, the example below includes logging the duplicate records to a LOG data set. Such a situation does not necessarily indicate a problem with the data set. If a record had to be moved from the block it was in into an overflow block, the overflow block is always immediately rewritten out to the data set. At a subsequent point in time, the original data block is rewritten with the record deleted. So, there is an opportunity for a record with the same key to be duplicated in the data set. IAM is able to handle this circumstance, and return the proper record. If there are duplicates, refer to Section 10.87, Recovering IAM Data Sets for complete instructions and examples of recovering files from that situation.

```
//RECOVER EXEC PGM=IAMRECVR,REGION=4M
//SYSPRINT DD   SYSOUT=*   
//SYSPUT   DD   SYSOUT-*   
//DISKIN   DD   DISP=OLD,DSN=my.iam.cluster
//TAPEOUT  DD   DSN=my.seq.dataset,DISP=(,CATLG),
//                UNIT=SYSDA,SPACE=(CYL,(20,10))
//LOG      DD   DSN=my.duprec.dataset,DISP=(,CATLG),
//                UNIT=SYSDA,SPACE=(CYL,(2,1))
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTPARM DD   *   -- Use for SyncSort
EQUALS
/*
//DFSPARM DD   *   -- Use for DFSORT
EQUALS
/*
//SYSIN DD   *
//RECOVER   DUP=LOG
/*
```

Figure 45-14: Example of Running a Recover (EX4509A)
Example B: Recover

The example below shows a RECOVER using the NOPRIMESORT option which causes two output files to be created, one containing all the records from the Prime Data Area and the other containing all records from Extended Overflow.

```
//RECOVER EXEC PGM=IAMRECVR
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//DISKIN DD DISP=SHR,DSN=my.iam.dataset 
//TAPEOUT DD DSN=&SEQOUT,DISP=(,PASS),SPACE=(CYL,(100,50)), 
//UNIT=SYSDA 
//OVERFLOW DD DSN=&OVOUT,DISP=(,PASS),SPACE=(CYL,(100,50)), 
//UNIT=SYSDA 
//SORTWK01 DD SPACE=(CYL,(10)),UNIT=SYSDA 
//SORTWK02 DD SPACE=(CYL,(10)),UNIT=SYSDA 
//SORTWK03 DD SPACE=(CYL,(10)),UNIT=SYSDA 
//SYSIN DD * 
RECOVER NOPRIMESORT
/*
**************************************************************************
//*        RELOAD IAM FILE WITH RECORDS FROM PRIMARY OUTPUT FILE
**************************************************************************
//RELOAD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=* 
//IAMFILE DD DISP=SHR,DSN=my.iam.file 
//INPUT DD DISP=(OLD,DELETE,DELETE),DSN=&SEQOUT 
//SYSIN DD * 
REPRO INFILE(INPUT) OUTFILE(IAMFILE) REUSE 
/*
**************************************************************************
//*        ADD RECORDS FROM OVERFLOW BACK TO NEW IAM FILE
**************************************************************************
//INSERTOV EXEC PGM=IAMRECVR
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//VSAMOUT DD DISP=SHR,DSN=IAMV.KSD428P.CLUSTER 
//OVERFLOW DD DSN=&OVOUT,DISP=(OLD,DELETE,DELETE) 
//SYSIN DD * 
APPLY LOGDDNAME=OVERFLOW 
/*
```

Figure 45-15: Example of using NOPRIMESORT (EX4509B)
46 IAMISPF – IAM’S ISPF INTERFACE

46.01 IAMISPF Overview

IAM includes an optional set of ISPF panels that can be used to perform many utility functions against IAM and most VSAM data sets. The IAM ISPF interface offers an interactive, fill in the blanks type of functionality for various utility functions. The panels also include a complete ISPF tutorial that describes how to use the panels, as well as a complete description of the key IAM OVERFLOW statistics and OVERRIDE parameters.

Data Set Naming Conventions

The following rules will apply in all of the IAM ISPF panels for specifying a data set name. To specify a fully qualified data set name, follow the TSO rules, which are to specify the data set name within apostrophes. If no apostrophes are provided, then the data set name will be automatically prefixed with the current TSO setting for data set prefixing, which is normally the TSO userid. If the provided data set name already has the current TSO prefix as the high level qualifier, then the prefix will not be added to the name.

Features

The key features included in the IAM ISPF panels are:

- Definition of IAM data sets with full Override support.
- Definition of IAM Alternate Indexes and Paths
- Definition of VSAM clusters.
- Definition of VSAM Alternate Indexes and Paths
- Definition using an IAM or VSAM model dataset.
- Multi-Volume dataset support.
- SMS support.
- Building IAM or VSAM Alternate Indexes
- Deletion of Data sets, Clusters, Paths, and Alternate Indexes.
- Renaming of Data sets, Clusters, Paths, and Alternate Indexes.
- Copy/Move support of IAM data sets, VSAM ESDS's and VSAM KSDS's.
- IAM or VSAM data sets can be copied into or from sequential data sets.
- Full IAM dataset information.
- VSAM cluster information.
- Interactive execution of selected IAM utility functions.

Users that are familiar with ISPF and that have knowledge of IAM or VSAM should find the IAM ISPF panels easy to use. Several examples of using the IAM ISPF panels are provided in the User’s Guide portion of the IAM manual.
IAMJREST is used to restore an IAM data set to its desired state using the journal records collected by the optional IAM journal exit. IAMJREST can use the IAM log data sets produced by non-RLS file updates or the IAM/RLS or IAM/PLEX log data sets. The IAM journal exit is activated by use of the IAM Override facility, by specifying the JRNAD keyword. IAM journaling is described in Section 10.88 IAM Journal and Recovery for non-RLS processing, in Section 20.30 IAM/RLS Journaling for IAM/RLS, and Section 25.30 IAM/PLEX Journaling for IAM/PLEX.

In the IAM/RLS and IAM/PLEX environment, IAMJREST, along with the locking services of IAM/RLS and IAM/PLEX, provides a recovery mechanism that can be used to back out updates made by an abending job step, while retaining the updates to other records made by concurrent jobs. IAMJREST can also be used as part of a forward recovery process, when an IAM data set had to be restored. In this case, IAMJREST can reapply all the updates to the restored file that occurred since it was backed up.

The use of these capabilities in a non-RLS environment can be helpful to improve data availability by reducing the frequency of backing up the complete data set, and enhancing recovery after a batch job abends. For example, rather than backing up the complete data set daily, the complete data set can be backed up less frequently, while backing up the IAM journal data set on a daily basis. This provides the same level of recovery capability, while reducing the backup time presuming that the amount of update activity is low to moderate. Recovery from batch job abends can also be improved by using the ability to back out the file updates that were done by the job step(s), as opposed to restoring the data set and rerunning file updates to get back to where the failure occurred.

The capabilities of IAMJREST include:

- Perform a forward recovery. After an IAM data set has been restored from a backup copy, IAMJREST can be used to apply all of the updates to the data set from the journal file.

- Perform a back out (or backwards) recovery. This type of recovery will back out changes from the specified job step(s), to restore a data set to an image prior to the start of the desired job or job step.

- Perform a back out recovery for multiple data sets that were being accessed under IAM/RLS or IAM/PLEX for failed batch jobs. If the failed batch job has performed IAM batch syncpoints, then the back out can optionally be performed to the most recent syncpoint prior to the abend.

- Release record locks being held without performing a recovery through IAMJREST.
A forward recovery is a process that will update a file with the updated records that are selected from the journal. The forward recovery process begins by independently restoring the file's contents from a backup copy. After a successful restore, the file is updated using the IAMJREST RESTORE FORWARD command with all of the updated records that are on the journal, up to the specified point in time. Actually, IAMJREST will sort the selected journal records by key (or RBA for ESDS files) and the time stamp. Then IAMJREST will only apply to the IAM file the most recent update for each record.

To be able to perform a forward recovery, the journal file must contain after images. This is accomplished by specifying either JRNAD=AFTER or JRNAD=BOTH on the IAM CREATE override. To be able to perform a forward recovery encompassing all of the file updates, then the IAM journal must be specified when the file is defined or loaded using the IAM CREATE override. This way there is not a concern about having missed specifying journaling on any of the update jobs.

For a typical journal set up to perform forward recoveries, the journal data set should be backed up and emptied at the same time of the data set backup. This will eliminate the need of identifying any start point for the forward recovery process, as is done by specifying any of the 'FROM' keywords on the restore command. There may be a need however to use the 'TO' keywords to identify an end point.

For example, let's say that there are seven nightly jobs that update a file, called A, B, C, D, E, F, and G. The situation we are setting up a recovery for is a media failure that occurred while job F was running for the third time this week. The job stream can be restarted with job E. We need to restore up to and including job D that ran on the third day. So, the RESTORE command would be:

```
RESTORE FORWARD,TOJOB=D,TODATE=day3
```

This would apply the updates for all seven jobs on the first two days, and up through and including job D on the third day. When completed, the data set is ready for rerunning job E, and moving forward.
Backout Recovery Process

The backout recovery is a process where the most recent updates to a data set are removed. This is accomplished by updating the data set using the before images of records, which are the images of records prior to being updated or deleted. Before images will also include images of inserted records, which will cause a record deletion by a backout recovery. A backout recovery is useful for restoring a data set to the contents it had prior to the start of job or job steps that updated the data set. While a forward recovery could be used if all of the after images have been collected, it will probably be faster just to back out the updates from the failed job rather than reapply what might be several days worth of updates.

To be able to perform a backout recovery process, the journal must contain the before images of the updated records. This is accomplished by specifying either JRNAD=BOTH or JRNAD=BEFORE on either a CREATE or ACCESS override. If you wanted to always be able to back out the updates from any job, then be sure to provide the specification on a CREATE override, just as you would with using a forward recovery. However, if you only wanted to provide backout ability for selected jobs, the ACCESS override could be specified for the jobs on which you wanted backout ability.

When setting up the control card for a backout recovery, generally you will want to back out the updates from one or more jobs and/or job steps that are at the end of the journal data set. So, you will not need to specify any of the ‘TO’ keywords to terminate selection, but you would specify some of the ‘FROM’ keywords, or perhaps just the JOBNAME and/or STEPNAME keywords. For example, if job G was the last update job to run, and it abended in one of the steps updating the IAM file. To back out all of the updates made by job G, the restore command would be:

RESTORE BACKOUT,JOBNAME=G

If the journal data set may contain multiple jobs by the name of G, the control statement would be changed to:

RESTORE BACKOUT,JOBNAME=G,FROMDATE=today,FROMTIME=lasttime

Another example would be if you have run jobs A, B, C, D, E, F, and G, with job G abending. For logistical reasons you need to restart with job E. The control statement would look like:

RESTORE BACKOUT,FROMJOB=E,FROMDATE=today

Command Summary

The IAMJREST program has the following commands / functions:

- **RESTORE**: This will invoke either the forward or back out recovery processing.
- **RELEASE**: This will release the record locks for the specified job without performing a recovery.

Please note that IAMJREST will only perform one command per execution.
47.02 IAMJREST – JCL Requirements

The JCL statements required to execute IAMJREST are as follows:

**Execute Statement**

Specifies the name of the IAM journal recovery / restore program – IAMJREST. For a file restore operation, sufficient storage on the REGION parameter must be specified to include storage for your system SORT. The SORT is invoked dynamically for sorting the selected records from the journal data set to expedite recovery processing. A typical execute statement would be:

```
//RESTORE EXEC PGM=IAMJREST,REGION=64M
```

**DD Statements**

The following table identifies the DD statements required by IAMJREST.

<table>
<thead>
<tr>
<th>DD Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB or JOBLIB</td>
<td>An optional DD statement that specifies the library containing the IAM program load modules. This DD is not necessary if IAM is in the Link List, as is recommended.</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>Required DD statement that specifies where the IAMJREST messages are to be printed. This is usually a SYSOUT data set.</td>
</tr>
<tr>
<td>SYSOUT</td>
<td>Required DD statement for RESTORE command that specifies the output message data set for the SORT. This is usually a SYSOUT data set.</td>
</tr>
<tr>
<td>IAMINFO</td>
<td>Optional DD statement that specifies the output report data set that contains the statistics for the IAM file activity that is performed by the IAMJREST program. This is usually a SYSOUT data set.</td>
</tr>
<tr>
<td>IAMJRNL</td>
<td>Optional DD statement that specifies the IAM journal data set(s) for the IAM file to be recovered. If the journal data set is backed up daily to a sequential file, the backup copies can be concatenated with the current journal data set. If such concatenation is done, be sure it is done such that the records will be read by IAMJREST in ascending time sequence. The oldest journal data must be first in the concatenation, and the most recent must be the last in the concatenation. If not specified, then IAMJREST will default to using the current IAM RLS log data sets to attempt to perform the requested recovery.</td>
</tr>
<tr>
<td>IAMFILE</td>
<td>Optional DD statement for RESTORE command from non-RLS journal that specifies the IAM file to be restored by IAMJREST. This DD is ignored for RESTORE commands from RLS journals.</td>
</tr>
<tr>
<td>SORTWKnn</td>
<td>These DD statements may be required when performing a RESTORE operation to utilize the system SORT. Be sure to provide adequate SORT work space (as specified on the SORTWKnn DD statements.)</td>
</tr>
<tr>
<td>SORTLIB</td>
<td></td>
</tr>
<tr>
<td>SYSIN</td>
<td>Required DD statement that contains the card image input of the command to be executed by IAMJREST. This is normally a sysin (DD *) type of data set.</td>
</tr>
</tbody>
</table>
IAMJREST – RESTORE Command

The RESTORE command will perform the indicated file recovery process. Optional record selection capability is provided based on the additional keywords. If processing a non-RLS journal, the RESTORE command will update the IAM data set specified on the IAMFILE DD card with the data record images from the IAM Journal data set specified on the IAMJRNL DD card. If processing from an IAM RLS journal, the RESTORE command will update all IAM data sets that meet the specified criteria, unless limited by the DSN keyword. Please note that updates performed by IAMJREST will not be written to the journal by the IAM journaling function.

When multiple selection criteria are specified, all of the indicated conditions must be met for the journal record to be selected for the recovery process.

```
RESTORE  FORWARD | BACKOUT

[,DSNAMES = IAM data set name]
[,EMPTYOK]        [,FROMDATE = yyyyddd]
[,FROMJOB = jobname] [,FROMSTEP = stepname]
[,FROMTIME = hhmmss] [,JOBDATE= yyyyddd]
[,JOBD=jes job id] [,JOBNAME = jobname]
[,JOBTIME=hhmmss]   [,JOURNAL = dsn.journal]
[,LOGSTREAMNAME=]  [,RLSID=cccc]
[,ROUTER=cccc]     [,SEQUENTIAL]
[,SIMULATE]        [,STEPNAME = stepname]
[,SYNCPOINT]       [,TARGET=cccc]
[,TODATE = yyyyddd] [,TOJOB = jobname]
[,TOSTEP = stepname] [,TOTIME = hhmmss]
[,TRANID= trans name] [,TRANNO=nnnnn]
```

Figure 47-1: IAMJREST RESTORE Command Operands
**RESTORE Command Operands**

**Operand** | **Description**
--- | ---
**FORWARD** | Specifies that a forward recovery will be performed. The forward recovery process is where the inserts, updates, and deletes from the selected journal records are performed on the IAM data set. A forward recovery is done after restoring an IAM data set, providing a mechanism to bring the file up to a specified point in time from the journal. To perform a forward recovery, the after images must be captured by the IAM journal, as indicated by specifying either JRNAD=BOTH or JRNAD=AFTER. Default is FORWARD recovery.

**BACKOUT** | Specifies that a backout (or backwards) recovery will be performed. The backout recovery process is where any updates, inserts, or deletes performed from the selected journal records will be removed from the file by updating the data set with the version of the record prior to the update. Records that were inserted will be deleted, and records that were deleted will be inserted. This process is accomplished by sorting the selected journal before images into a descending time sequence. To perform a back out recovery, the IAM journal must have before images, as indicated by specifying either JRNAD=BOTH or JRNAD=BEFORE. Default is FORWARD recovery.

**DSNAMES=** | Specifies the IAM data set for which recovery is to be performed. Default for non-RLS journal file input is the data set identified by the IAMFILE DD statement. For IAM/RLS of IAM/PLEX processing, the default is all IAM files processed for the selected journal records.

**EMPTYOK** | Optional keyword that sets the return code to 0 even if no records were selected due to an empty journal.

**FROMDATE=** | Specifies the lower time limit for journal records to be included in the recovery process. All preceding records are ignored. Must be specified in the form of ‘yyyyddd’, where yyyy is the 4-digit year value, and ddd is the 3-digit julian day value. Default is none, the date is not used for starting journal record selection.

**FROMJOB=** | A 1 to 8 character value for jobname that specifies that journal records will be selected for processing starting with the first journal record found for the specified jobname. All preceding journal records are ignored. Default is none, the jobname is not used for starting journal record selection.

**FROMSTEP=** | A 1 to 8 character value for step name that specifies that journal records will be selected for processing starting with the first journal record found for the specified step name. All preceding journal records are ignored. Default is none, the step name is not used for starting journal record selection.

**FROMTIME=** | Specifies that journal records will be selected for processing starting with the first journal record found for the specified time. All preceding journal records are ignored. The time is specified as either a 4 character ‘hhmm’ or a 6 character ‘hhmmss’ value, using an ‘hh’ value based on a 24-hour clock. Default is none, the time will not be used for starting journal record selection.
<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBDATE=</td>
<td>For a back out recovery from an IAM RLS journal, specifies the start date of the job from where recovery is to start. Additionally, when specified for an execution where IAMJREST is selecting from the IAM RLS journal data sets, IAMJREST will find the journal that will have the data for this date. Default is no date criteria will be used, and when automatic selection of the IAM RLS journal is being done, IAMJREST will use the currently active IAM RLS journal.</td>
</tr>
<tr>
<td>JOBID=</td>
<td>For a back out recovery from the IAM RLS journal data sets, specifies the JES JOBID in the format of JOB##### or J####### as applicable for your installation. Default is that the JOBID is not used for selection. For a back out recovery from the IAM RLS journal, either a JOBNAME and / or JOBID must be specified.</td>
</tr>
<tr>
<td>JOBNAME=</td>
<td>A 1 to 8 character value specifying that only those journal records with a matching jobname will be selected for the recovery process. A special value of * can be used, which indicates the jobname of the job executing the IAMJREST recovery. Default is none, there is no jobname restriction on journal record selection.</td>
</tr>
<tr>
<td>JOBTIME=</td>
<td>For a back out recovery from the IAM RLS journals, specifies the starting time of the job for which the back out is to occur. This value will also be used by IAMJREST when selecting which IAM RLS journal to use. Default is that job start time is not used for selection.</td>
</tr>
<tr>
<td>JOURNAL=</td>
<td>Optional keyword to indicate which journal file is to be used during the restore request. This is required if the journal is currently active by IAM/RLS or IAM/PLEX and can not be specified in JCL. Doing so would cause the job to wait for the specified journal to be free.</td>
</tr>
<tr>
<td>LOGSTREAMNAME=</td>
<td>Optional keyword that indicates that IAMJREST will use the specified MVS system logger stream for the recovery process. The LOGSTREAMNAME is only applicable for IAM RLS journals. Default is that the standard sequential log data sets will be used.</td>
</tr>
<tr>
<td>RLSID=ccccc</td>
<td>Optional keyword that is used when recovering from active IAM/RLS or IAM/PLEX journals, the four character identifier of the address space that has the journals to be recovered from. Default is to use the RLSID from the IAM Global Options Table.</td>
</tr>
<tr>
<td>ROUTER=ccccc</td>
<td>Optional keyword to indicate that only journal records with the specified IAM RLS ROUTER RLSID should be included in the restore. Default is ROUTER RLSID will not be used during record selection.</td>
</tr>
<tr>
<td>SEQUENTIAL</td>
<td>Optional keyword that indicates that during a FORWARD recovery, all updated images will be applied as they occur in the journal. Default is that only the last image of a particular record as identified by key or RBA will be used to update the file to save processing time.</td>
</tr>
<tr>
<td>SIMULATE</td>
<td>Optional keyword that indicates that the RESTORE only be simulated so the results of the request can be seen before the actual RESTORE is done.</td>
</tr>
<tr>
<td><strong>Operand</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>STEPNAME=</strong></td>
<td>A 1 to 8 character value specifying that only those journal records with a matching step name will be selected for the recovery process. Default is none, there is no step name restriction on journal record selection.</td>
</tr>
<tr>
<td><strong>SYNCPOINT</strong></td>
<td>An optional keyword for back out recoveries of batch jobs that indicate the recovery is performed back to the most recent syncpoint. If no batch syncpoint is found, the back out will be for all of the records from the start of the job step. Default is that SYNCPOINT records are ignored during back out recovery.</td>
</tr>
<tr>
<td><strong>TARGET=cccc</strong></td>
<td>Optional keyword to indicate that only journal records with the specified IAM RLS TARGET RLSID should be included in the restore. Default is TARGET RLSID will not be used for record selection.</td>
</tr>
<tr>
<td><strong>TODATE=</strong></td>
<td>Specifies the journal records up to and including the specified date will be eligible for the recovery process. Must be specified in the form of ‘yyyyddd’, where yyyy is the 4-digit year value, and ddd is the 3-digit julian day value. Default is none, there is no date limitation on journal record selection.</td>
</tr>
<tr>
<td><strong>TOJOB=</strong></td>
<td>A 1 to 8 character value specifying that journal records up to and including the specified job will be eligible for the recovery process. All journal records after those for the specified job are ignored. Default is none, there is no jobname value for terminating journal record selection.</td>
</tr>
<tr>
<td><strong>TOSTEP=</strong></td>
<td>A 1 to 8 character value specifying that journal records up to and including the specified step will be eligible for the recovery process. All journal records after those for the specified step are ignored. Default is none, there is no step name value for terminating journal record selection.</td>
</tr>
<tr>
<td><strong>TOTIME=</strong></td>
<td>A time value in the form of either a 4 character ‘hhmm’ or 6 character ‘hhmms’ specifying that all journal records up to and including the specified time value are eligible for the recovery process. Journal records with time values higher than the specified time value will be ignored. Default is none, there is no upper time limit for terminating journal record selection.</td>
</tr>
</tbody>
</table>
47.04 IAMJREST Examples

A few examples of running IAMJREST are shown below. Additional examples, along with information on setting up and using the IAM Journaling facility are provided in Section 10-88 IAM Journal and Recovery.

Example A: Forward Restore Example
This example will show all the JCL and the control card to run the forward recovery that was discussed under Forward Recovery in Section 47.01. For this example, we are recovering up to and including all the updates done by 'jobD' that was run on October 22, 2014 (julian 2014.295), that had ended by 11:15pm.

```plaintext
//FORWARD EXEC PGM=IAMJREST,REGION=64M
//SYSPRINT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//IAMINFO DD SYSOUT=*  
//IAMJRNL DD DSN=my.iam.ksd.log,DISP=OLD 
//IAMFILE DD DSN=my.iam.ksd,DISP=OLD 
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50,10))  
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50,10))  
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50,10))  
//SYSIN DD *  
   RESTORE FORWARD,TOJOB=jobD,TODATE=2014295,TOTIME=2315 */
```

Figure 47-2: Example of Forward Recovery with TOJOB, TODATE, and TOTIME. (EX4704A)

Example B: back out Restore
This example will show all the JCL and the control card to run a back out recovery. The circumstance is that we want to back out the updates done by job UPDG, which has been run several times. We just want to back out the updates from the last run, which started at 10:30am.

```plaintext
//back out EXEC PGM=IAMJREST,REGION=64M
//SYSPRINT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//IAMINFO DD SYSOUT=*  
//IAMJRNL DD DSN=my.iam.ksd.log,DISP=OLD 
//IAMFILE DD DSN=my.iam.ksd,DISP=OLD 
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50,10))  
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50,10))  
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50,10))  
//SYSIN DD *  
   RESTORE back out,FROMJOB=UPDG, FROMDATE=2014295,FROMTIME=1030 */
```

Figure 47-3: Example of a back out Restore (EX4704B)
This example will show the JCL to run a back out recovery from within an abending job that was accessing IAM data sets with IAM RLS. To automate the recovery, the IAMJREST job step will be invoked only if a preceding job step has abended. The back out will be performed from the last IAM batch SYNCPOINT that was performed prior to the abend. This recovery will use the current IAM RLS log data set(s).

```java
//back out EXEC PGM=IAMJREST,COND=ONLY,REGION=0M
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//IAMINFO DD SYSOUT=* 
//SYSIN DD * 
    RESTORE back out,SYNCPOINT,JOBNAME=* 
/*
```

Figure 47-4: Example of an IAM RLS back out Restore (EX4704C)
**IAMJREST – RELEASE Command**

The RELEASE command will cause retained record locks that are being held by IAM/RLS or IAM/PLEX to be released. It is similar in function to the IAM/RLS and IAM/PLEX Released Locks command. This process will only release record locks for one unit of work, that is one jobstep or one CICS transaction. Specify enough information to uniquely identify the unit of work that the records locks are to be released for. If there are multiple matches found, no record locks will be released. Information on jobs or transactions that have retained record locks can be found by issuing the IAM/RLS or IAM/PLEX Display Retained Locks command.

```
RELEASE
   [,ASID= xxxx] [,JOBID=jes jobid]
   [,JOBNAME = jobname] [,RLSID=cccc]
   [,STEPNAME = stepname ] [,TRANID= trans name]
   [,TRANNO=nnnnn]
```

*Figure 47-5: IAMJREST RELEASE Command Operands*

<table>
<thead>
<tr>
<th><strong>Operand</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ASID=</td>
<td>Specifies the address space identifier (ASID) of the job that the record locks are to be released for. Value is specified in hexadecimal digits.</td>
</tr>
<tr>
<td>Default is no ASID will be used to find the job for which the locks are to be released.</td>
<td></td>
</tr>
<tr>
<td>JOBID=</td>
<td>Specifies the JES JOBID in the format of JOB##### or J####### as applicable for your installation.</td>
</tr>
<tr>
<td>Default is that the JOBID is not used for selection to find the job whose locks are to be released.</td>
<td></td>
</tr>
<tr>
<td>JOBNAME=</td>
<td>Specifies a 1 to 8 character value of the name of the job whose record locks are to be released.</td>
</tr>
<tr>
<td>A special value of * can be used, which indicates the jobname of the job executing the IAMJREST recovery.</td>
<td></td>
</tr>
<tr>
<td>Default is none, the jobname will not be used to find the job for which the record locks are to be released.</td>
<td></td>
</tr>
<tr>
<td>RLSID=cccc</td>
<td>Optional keyword that is used when recovering from active IAM/RLS or IAM/PLEX journals, the four character identifier of the address space that handled the job whose locks are to be released. For IAM/PLEX this would be known as the ROUTER IAM/PLEX address space.</td>
</tr>
<tr>
<td>Default is to use the RLSID from the IAM Global Options Table.</td>
<td></td>
</tr>
<tr>
<td><strong>Operand</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>STEPNAME=</strong></td>
<td>A 1 to 8 character value specifying the name of the job step whose record locks are to be released. Default is none, the step name will not be used to find the job whose record locks are to be released.</td>
</tr>
<tr>
<td><strong>TRANID</strong></td>
<td>Specifies the 1 to 4 character CICS transaction name or identifier of the CICS transaction whose record locks are to be released.</td>
</tr>
<tr>
<td><strong>TRANNO</strong></td>
<td>Specifies the 1 to 5 digit CICS transaction number whose record locks are to be released.</td>
</tr>
</tbody>
</table>
IAMJUTIL Program Overview

IAMJUTIL is a utility program that provides functions to assist in managing the IAM journals. These capabilities include:

- **COPY** function that will copy all or selected records from the journal file to another data set. This can be used to offload or backup the journal file(s).
- **DUMP** function that can be used to print off all or selected records from an IAM journal.
- The **MVSLOG** function to remove old data from the MVS System Logger being used by IAM RLS.
- **SCAN** function, that will produce a summary report of the data contained on a journal file, to indicate what job(s) and step(s) have data in the journal.
48.02 IAMJUTIL – JCL Requirements

The JCL statements required to execute IAMJUTIL are as follows:

**EXEC Statement**

Specifies the IAM Journal Utilities program name – IAMJUTIL.

**DD Statements**

The following table describes the required DD statements for running IAMJUTIL.

<table>
<thead>
<tr>
<th>DD Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB or JOBLIB</td>
<td>Specifies the IAM Load Module Library. This DD statement may be omitted if the IAM load library is in the link list.</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>Required DD statement that specifies where the printed output is to go, normally it is a SYSPUT=* dataset.</td>
</tr>
<tr>
<td>SYSIN</td>
<td>Required DD statement that specifies the control card input dataset, normally a DD * dataset.</td>
</tr>
<tr>
<td>IAMJRNL</td>
<td>Required DD statement that specifies the input IAM journal datasets.</td>
</tr>
<tr>
<td>JRNLOUT</td>
<td>Specifies the output journal file when a COPY operation is being performed. This is a sequential output disk or tape file, with DCB characteristics of the input journal dataset.</td>
</tr>
</tbody>
</table>

*Figure 48-1: DD Statements Required for IAMJUTIL*
The COPY command is used to copy selected IAM Journal file records to another file for backup or other purposes. The selection operands act in a logical AND manner. A journal record has to meet all of the selection criteria to be eligible to be copied. To determine the contents of the IAM journal file, use the SCAN command.

**COPY Operands**

The following operands can be specified on the COPY command. The underscored portion indicates the minimum abbreviation that can be specified for the keyword.

<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFTER</td>
<td>Optional keyword that specifies IAMJUTIL is to copy only the journal records containing after images. Suggested use is when offloading an IAM RLS journal from which there will be no future backout recoveries performed, copying only the after images will reduce the size of the offloaded dataset. After images are used for forward recoveries. Default is that both before and after images are eligible to be copied.</td>
</tr>
<tr>
<td>COUNT=</td>
<td>Optional keyword that specifies IAMJUTIL is to copy only the journal records containing before images. Before images are only used for backout recoveries. Default is that both before and after images are eligible to be copied.</td>
</tr>
<tr>
<td>DATAONLY</td>
<td></td>
</tr>
<tr>
<td>EMPTYOK</td>
<td></td>
</tr>
<tr>
<td>FORCEREUSE</td>
<td></td>
</tr>
<tr>
<td>FROMBLOCKID</td>
<td></td>
</tr>
<tr>
<td>FROMBLOCKNO</td>
<td></td>
</tr>
<tr>
<td>FROMDATE</td>
<td></td>
</tr>
<tr>
<td>FROMTIME</td>
<td></td>
</tr>
<tr>
<td>FORTRANCK</td>
<td></td>
</tr>
<tr>
<td>GENERICKEY</td>
<td></td>
</tr>
<tr>
<td>IGNOREREUSE</td>
<td></td>
</tr>
<tr>
<td>JOBID</td>
<td></td>
</tr>
<tr>
<td>JOBNAME</td>
<td></td>
</tr>
<tr>
<td>JOURNAL</td>
<td></td>
</tr>
<tr>
<td>LOGSTREAMNAME</td>
<td></td>
</tr>
<tr>
<td>OUTDD</td>
<td></td>
</tr>
<tr>
<td>REUSE</td>
<td></td>
</tr>
<tr>
<td>ROUTER</td>
<td></td>
</tr>
<tr>
<td>STEPNAME</td>
<td></td>
</tr>
<tr>
<td>TOBLOCKID</td>
<td></td>
</tr>
<tr>
<td>TOBLOCKNO</td>
<td></td>
</tr>
<tr>
<td>TODATE</td>
<td></td>
</tr>
<tr>
<td>TOTIME</td>
<td></td>
</tr>
<tr>
<td>TRANID</td>
<td></td>
</tr>
<tr>
<td>TRANNO</td>
<td></td>
</tr>
<tr>
<td><strong>Operand</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
</tr>
<tr>
<td>DATAONLY</td>
<td>Optional keyword that specifies IAMJUTIL is to copy only the IAM data records to the output file. The journal record heading information will be eliminated. Default is that the entire journal record will be copied, including the descriptive header information.</td>
</tr>
<tr>
<td>DSN=</td>
<td>Optional keyword that specifies IAMJUTIL is to copy from an IAM RLS journal only those journal records for the specified IAM dataset. Default is that the journal records for all of the IAM data sets will be copied.</td>
</tr>
<tr>
<td>EMPTYOK</td>
<td>Optional keyword that sets the return code to 0 even if no records were copied due to an empty journal.</td>
</tr>
<tr>
<td>FORCEREUSE</td>
<td>Optional keyword to force the setting of the indicator in the journal file that the file can be reused when required.</td>
</tr>
<tr>
<td>FROMBLOCKID=</td>
<td>Optional keyword that specifies IAMJUTIL is to copy journal records from the specified MVS system logger dataset starting with the specified block id number. Default is to start with the first record in the MVS system logger.</td>
</tr>
<tr>
<td>FROMBLOCKNO=</td>
<td>Optional keyword that specifies IAMJUTIL is to copy journal records from a sequential IAM journal file starting from the specified relative block number. Default is to start with the first block in the input journal file.</td>
</tr>
<tr>
<td>FROMDATE=</td>
<td>Optional keyword that specifies IAMJUTIL will copy records starting with those that have the specified date. The date is specified as a julian date in the format yyyyddd. Default is that the copy will begin with the first record in the journal file, regardless of the date.</td>
</tr>
<tr>
<td>FROMTIME=</td>
<td>Optional keyword that specifies IAMJUTIL will copy records starting with those that have the specified time. The format of the time, which is based on 24-hour period, is hhmmsssth. Default is that the copy will begin with the first record in the journal file, regardless of the time.</td>
</tr>
<tr>
<td>FROMTTR=tttttttrr</td>
<td>Optional keyword to indicate that only journal records with a TTR of this value or higher are to be copied. The TTR value can be obtained from IAMJUTIL SCAN or DUMP output. The input value is hex digits with up to 6 digits for the track, and 2 digits for the record number field.</td>
</tr>
</tbody>
</table>
### IAMJUTIL – COPY Command

<table>
<thead>
<tr>
<th><strong>Operand</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GENERICKEY</strong></td>
<td>Optional keyword to indicate that only records with the same generic key should be copied. This option will cause the key in the record to be matched up to the number of bytes specified. If equal, the record will be copied. The actual key values must be entered using the KEY1 through KEY8 parameters described below.</td>
</tr>
<tr>
<td><strong>IGNOREREUSE</strong></td>
<td>Optional keyword to indicate that no error message is to be issued if the REUSE bit is already set in the journal file being processed.</td>
</tr>
<tr>
<td><strong>JLOG</strong></td>
<td>Optional keyword that specifies IAMJUTIL will produce a printed log of all activity performed by the copy command. Default is no detailed log will be produced.</td>
</tr>
<tr>
<td><strong>JOBID=</strong></td>
<td>Optional keyword that specifies IAMJUTIL will copy records for the JES JOBID (usually JOBnnnnn or Jnnnnnnn) specified. Default is that records for all jobs are eligible to be copied.</td>
</tr>
<tr>
<td><strong>JOBNAME=</strong></td>
<td>Optional keyword that specifies IAMJUTIL will copy only the journal records that have a matching jobname. If multiple jobs with the same name are encountered, then all of the jobs with a matching name will be copied, unless restricted by other operands. Default is that journal records for all jobs are eligible to be copied.</td>
</tr>
<tr>
<td><strong>JOURNAL=dsname</strong></td>
<td>Optional keyword to indicate which dataset name of the journal file is that is to be used during the COPY request. This is required if the journal is currently active by IAM/RLS or IAM/PLEX and can not be specified in JCL. Doing so would cause the job to wait for the specified journal to be free.</td>
</tr>
<tr>
<td><strong>KEY1=xxxxxxxx</strong></td>
<td>Optional keywords to enter up to 512 hex digits to create up to 256 byte key to be used during the copy. Each KEYx keyword can hold up to 64 hex characters which will be converted to 32 bytes. All entered values will be combined to make a single key (or generic key) to be used during the COPY.</td>
</tr>
<tr>
<td><strong>KEY2=xxxxxxxx</strong></td>
<td></td>
</tr>
<tr>
<td><strong>KEY8=xxxxxxxx</strong></td>
<td></td>
</tr>
<tr>
<td><strong>LOGSTREAMNAME=</strong></td>
<td>Specifies the name of the MVS system logger dataset from which IAMJUTIL will copy records. This keyword is required when processing an IAM RLS system logger journal dataset.</td>
</tr>
<tr>
<td><strong>JRNLOUT</strong></td>
<td>Default is that IAMJUTIL assumes the copy is from a sequential journal dataset.</td>
</tr>
<tr>
<td><strong>OUTDD=</strong></td>
<td>Optional keyword that specifies the name of the DD statement that defines the dataset into which the data will be copied. Default is JRNLOUT.</td>
</tr>
<tr>
<td>Operand</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>REUSE</td>
<td>Optional keyword that can be used when copying data from an IAM RLS journal file. This will set an indication to IAM RLS that the journal file can be reused when needed, but in the interim period the data in the journal remains available for automated backout processing. This is recommended when offloading an active IAM RLS journal. Default is that IAMJUTIL will not set the reusability indicator for this journal dataset.</td>
</tr>
<tr>
<td>RLSID=</td>
<td>Optional keyword to indicate that only journal records with the specified IAM RLSID should be included in the copy.</td>
</tr>
<tr>
<td>ROUTER=cccc</td>
<td>Select journal records based on RLSID of the ROUTER IAM/PLEX region.</td>
</tr>
<tr>
<td>Default is ROUTER RLSID will not be used during record selection.</td>
<td></td>
</tr>
<tr>
<td>STEPNAME=</td>
<td>Optional keyword that specifies that IAMJUTIL will copy only those records with the specified stepname.</td>
</tr>
<tr>
<td>Default is that journal records are eligible to be copied regardless of the stepname.</td>
<td></td>
</tr>
<tr>
<td>TARGET=cccc</td>
<td>Optional keyword to indicate that only journal records with the specified IAM/PLEX TARGET RLSID should be included in the copy.</td>
</tr>
<tr>
<td>Default is TARGET RLSID will not be used during record selection.</td>
<td></td>
</tr>
<tr>
<td>TOBLOCKID=</td>
<td>Optional keyword that can be used when copying journal data from an MVS system logger dataset, that indicates IAMJUTIL will copy only data up to and including the specified system logger record ID.</td>
</tr>
<tr>
<td>Default is that copying will end with the last record that is currently in the MVS system logger dataset.</td>
<td></td>
</tr>
<tr>
<td>TOBLOCKNO=</td>
<td>Optional keyword when copying from a sequential IAM journal, that specifies IAMJUTIL is to copy all data up that is contained in the blocks up to and including the specified relative block number.</td>
</tr>
<tr>
<td>Default is that copying will end with the last record that is in the sequential journal dataset.</td>
<td></td>
</tr>
<tr>
<td>TODATE=</td>
<td>Optional keyword that specifies IAMJUTIL is to copy all journal records up to and including the specified date. The date is specified as a julian date in the format yyyyddd.</td>
</tr>
<tr>
<td>Default is that copying will end with the last record found in the input journal dataset(s).</td>
<td></td>
</tr>
</tbody>
</table>
Example A: Using IAMJUTIL COPY to offload data

This is an example of how to use IAMJUTIL to offload data from an IAM RLS journal into an accumulation dataset, and mark the IAM RLS journal as being ready for reuse. Only the after images are being copied, because the journal data will be used only to perform a forward recovery from a restored datasets. Backout recoveries will not be possible from the accumulation dataset. Backout recoveries can still be performed from the IAM RLS journal until it is actually reused by IAM RLS.

```
//EX4803A  EXEC PGM=IAMJUTIL
//SYSUDUMP DD  SYSOUT=*  
//SYSPRINT DD  SYSOUT=*  
//SYSOUT   DD  SYSOUT=*  
//IAMJRNL  DD  DISP=SHR,DSN=PROD.IAMRLS.JOURNAL.LOGDSN2  
//JRNLOUT  DD  DISP=MOD,DSN=PROD.IAMRLS.OFFLOAD.JOURNAL  
//SYSIN    DD  *  
COPY AFTER,OUTDD=JRNLOUT,REUSE  
/*
```

Figure 48-3: Example of Offloading data from an IAM RLS Journal (EX4803A)
IAMJUTIL – DUMP Command

The DUMP command is used to produce a printed dump of selected IAM Journal file records. This may be useful if there are any problems when performing RESTORE functions with IAMJREST. The selection operands act in a logical AND manner. A journal record has to meet all of the selection criteria to be eligible to be dumped. To determine what jobs and steps have written any journal records, use the IAMJUTIL SCAN command.

DUMP Operands

The following operands can be specified on the DUMP command. The underscored portion indicates the minimum abbreviation that can be specified for the keyword.

<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>Optional keyword that specifies IAMJUTIL is to dump the entire contents of each journal record, including the header information and the actual data record. Default is that only the data portion is printed.</td>
</tr>
<tr>
<td>BLOCKS</td>
<td>Optional keyword that specifies IAMJUTIL is to dump the journal dataset by block, not by records. Default is to dump each journal record.</td>
</tr>
<tr>
<td>COUNT=</td>
<td>Optional keyword that specifies IAMJUTIL is to dump only the specified number of records. Default is that all journal records are eligible.</td>
</tr>
</tbody>
</table>

Figure 48-4: Figure 95: IAMJUTIL DUMP Command Operands
<table>
<thead>
<tr>
<th><strong>Operand</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>DSN=</td>
<td>Optional keyword that specifies IAMJUTIL is to dump from an IAM RLS journal only those journal records for the specified IAM dataset. Default is that the journal records for all of the IAM data sets will be dumped.</td>
</tr>
<tr>
<td>EMPTYOK</td>
<td>Optional keyword that sets the return code to 0 even if no records were dumped due to an empty journal.</td>
</tr>
<tr>
<td>FROMBLOCKID=</td>
<td>Optional keyword that specifies IAMJUTIL is to dump journal records from the specified MVS system logger dataset starting with the specified block id number. Default is to start with the first record in the MVS system logger.</td>
</tr>
<tr>
<td>FROMBLOCKNO=</td>
<td>Optional keyword that specifies IAMJUTIL is to dump the journal records from a sequential IAM journal file starting from the specified relative block number. Default is to start with the first block in the input journal file.</td>
</tr>
<tr>
<td>FROMDATE=</td>
<td>Optional keyword that specifies IAMJUTIL will dump records starting with those that have the specified date. The date is specified as a julian date in the format yyyyddd. Default is that the dump will begin with the first record in the journal file, regardless of the date.</td>
</tr>
<tr>
<td>FROMTIME=</td>
<td>Optional keyword that specifies IAMJUTIL will dump records starting with those that have the specified time. The format of the time, which is based on 24-hour period, is hhmmstth. Default is that the dump will begin with the first record in the journal file, regardless of the time.</td>
</tr>
<tr>
<td>FROMTTR=tttttttrr</td>
<td>Optional keyword to indicate that only journal records with a TTR of this value or higher are to be dumped. The TTR value can be obtained from IAMJUTIL SCAN or DUMP output. The input value is hex digits with up to 6 digits for the track, and 2 digits for the record number field.</td>
</tr>
<tr>
<td>GENERICKEY</td>
<td>Optional keyword to indicate that only records with the same generic key should be dumped. This option will cause the key in the record to be matched up to the number of bytes specified. If equal, the record will be copied. The actual key values must be entered using the KEY1 through KEY8 parameters described below.</td>
</tr>
<tr>
<td>HEADERONLY</td>
<td>Optional keyword that specifies IAMJUTIL will dump only the header portion of each journal record. Default is that IAMJUTIL will dump only the data portion of each journal record.</td>
</tr>
<tr>
<td><strong>Operand</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
</tr>
<tr>
<td><strong>JLOG</strong></td>
<td>Optional keyword that specifies IAMJUTIL will produce a printed log of all activity performed by the dump command. Default is no detailed log will be produced.</td>
</tr>
<tr>
<td><strong>JOBID=</strong></td>
<td>Optional keyword that specifies IAMJUTIL will dump records for the JES JOBID (usually JOBnnnnn or Jnnnnnnn) specified. Default is that records for all jobs are eligible to be dumped.</td>
</tr>
<tr>
<td><strong>JOBNAME=</strong></td>
<td>Optional keyword that specifies IAMJUTIL will dump only the journal records that have a matching jobname. If multiple jobs with the same name are encountered, then all of the jobs with a matching name will be dumped, unless restricted by other operands. Default is that journal records for all jobs are eligible to be dumped.</td>
</tr>
<tr>
<td><strong>JOURNAL=dsname</strong></td>
<td>Optional keyword to indicate which dataset name of the journal file is that is to be used during the DUMP request. This is required if the journal is currently active by IAM/RLS or IAM/PLEX and can not be specified in JCL. Doing so would cause the job to wait for the specified journal to be free.</td>
</tr>
<tr>
<td><strong>KEY1=xxxxxxxx</strong></td>
<td>Optional keywords to enter up to 512 hex digits to create up to 256 byte key to be used during the dump. Each KEYx keyword can hold up to 64 hex characters which will be converted to 32 bytes. All entered values will be combined to make a single key (or generic key) to be used during the DUMP.</td>
</tr>
<tr>
<td><strong>LOGSTREAMNAME=</strong></td>
<td>Specifies the name of the MVS system logger dataset from which IAMJUTIL will dump records. This keyword is required when processing an IAM RLS system logger journal dataset. Default is that IAMJUTIL assumes the dump is from a sequential journal dataset.</td>
</tr>
<tr>
<td><strong>RLSID=cccc</strong></td>
<td>Optional keyword to indicate that only journal records with the specified IAM RLSID should be dumped.</td>
</tr>
<tr>
<td><strong>ROUTER=cccc</strong></td>
<td>Select journal records based on RLSID of the ROUTER IAM/PLEX region. Default is ROUTER RLSID will not be used during record selection.</td>
</tr>
<tr>
<td><strong>STEPNAME=</strong></td>
<td>Optional keyword that specifies that IAMJUTIL will dump only those records with the specified stepname. Default is that journal records are eligible to be dumped regardless of the stepname.</td>
</tr>
<tr>
<td><strong>TARGET=cccc</strong></td>
<td>Optional keyword to indicate that only journal records with the specified IAM/PLEX TARGET RLSID should be dumped. Default is TARGET RLSID will not be used during record selection.</td>
</tr>
<tr>
<td><strong>Operand</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TOBLOCKID=</td>
<td>Optional keyword that can be used when dumping journal data from an MVS system logger dataset, that indicates IAMJUTIL will dump only data up to and including the specified system logger record ID.</td>
</tr>
<tr>
<td></td>
<td>Default is that dumping will end with the last record that is currently in the MVS system logger dataset.</td>
</tr>
<tr>
<td></td>
<td>Optional keyword when dumping from a sequential IAM journal, that specifies IAMJUTIL is to dump all data up that is contained in the blocks up to and including the specified relative block number.</td>
</tr>
<tr>
<td>TOBLOCKNO=</td>
<td>Default is that dumping will end with the last record that is in the sequential journal dataset.</td>
</tr>
<tr>
<td></td>
<td>Optional keyword that specifies IAMJUTIL is to dump all journal records up to and including the specified date. The date is specified as a julian date in the format yyyyddd.</td>
</tr>
<tr>
<td>TODATE=</td>
<td>Default is that dumping will end with the last record found in the input journal dataset(s).</td>
</tr>
<tr>
<td></td>
<td>Optional keyword that specifies IAMJUTIL is to dump all journal records up to and including the specified time. The time is specified in the format of hhmmsssth.</td>
</tr>
<tr>
<td>TOTIME=</td>
<td>Default is that dumping will end with the last record found in the input journal dataset(s).</td>
</tr>
<tr>
<td></td>
<td>Optional keyword that specifies IAMJUTIL will dump only those records from an IAM RLS journal for the specified CICS transaction name. TRANID specifies the 1 to 4 character transaction name.</td>
</tr>
<tr>
<td>TRANID=</td>
<td>Default is that all transactions are eligible to be dumped.</td>
</tr>
<tr>
<td></td>
<td>Optional keyword that specifies IAMJUTIL will dump only those records with the specified CICS transaction number. TRANNO specifies a 1 to5 digit transaction number.</td>
</tr>
<tr>
<td>TRANNO=</td>
<td>Default is that all transactions are eligible to be dumped.</td>
</tr>
<tr>
<td></td>
<td>Optional keyword to indicate that only journal records of the specified type should be included in the DUMP. Only one type can be specified. Valid values are one of the following: GETUPD, PUTUPD, PUTADD, DELETE, SYNC, CLOSE, STEPTERM or RLS</td>
</tr>
<tr>
<td>TYPE=</td>
<td>Default is all types will be dumped.</td>
</tr>
</tbody>
</table>
Example A:
Printing (DUMPING)
Selected Journal Records

In this example of the DUMP command, records for the specified job step will be printed.

```plain
//EX4804A  EXEC PGM=IAMJUTIL
//SYSUDUMP DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//IAMJRNL DD DISP=SHR, DSN=GFM.LOGDSN2.JOURNAL  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
/*
DUMP ALL, JOB=jobname, STEP=stepname, JOBID=JOBnnnnn
/*
```

Figure 48-5: Example of Dumping Selected Records (EX4804A)
IAMJUTIL – MVSLOG Command

The IAMJUTIL MVSLOG command is provided to help administer the MVS logstream, if users have specified the use of the logstream by IAM RLS. This command provides a mechanism to remove old records from the logstream, providing such is not done automatically or by other means.

```
MVSLOG  DELETE
    [,ALL]      [,BLOCKID=nnnnnnn]
    [,DATE=yyyddd]    [,LOGSTREAMNAME=logdsn]
    [,RLSID=cccc]    [,TIME=hhmmssth]
```

**Figure 48-6: IAMJUTIL MVSLOG Operands**

The following describes the operands that can be specified on the MVSLOG command. When an abbreviation is allowed, the underscored portion of the keyword represents the minimum allowable abbreviation. In addition to the required operand of DELETE, additional operands must be specified to indicate what data is to be removed from the log stream.

<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETE</td>
<td>Required keyword that specifies IAMJUTIL is to perform a record deletion request on the specified MVS log stream.</td>
</tr>
<tr>
<td>ALL</td>
<td>Optional keyword that specifies that IAMJUTIL is to request the deletion of all of the records in the specified log stream. Default is none, either ALL, BLOCKID or DATE must be specified.</td>
</tr>
<tr>
<td>BLOCKID=</td>
<td>Optional keyword that specifies that IAMJUTIL is to request deletion of all records preceding the specified block id. Default is none, either ALL, BLOCKID or DATE must be specified.</td>
</tr>
<tr>
<td>DATE=yyyddd</td>
<td>Optional keyword that specifies that IAMJUTIL is to request deletion of all records preceding the specified date, which must be specified as a julian date. Default is none, either ALL, BLOCKID or DATE must be specified.</td>
</tr>
<tr>
<td>LOGSTREAMNAME=</td>
<td>Required keyword that specifies the name of the MVS System Logger log stream that was used by IAM RLS.</td>
</tr>
<tr>
<td>RLSID=</td>
<td>Optional keyword to indicate that only blocks containing the specified RLSID should be deleted.</td>
</tr>
<tr>
<td>TIME=hhmmssth</td>
<td>Option keyword that can be specified with the DATE operand, that will specify to IAMJUTIL to request deletion of all records in the log stream preceding the specified date and time. Default is 00000000.</td>
</tr>
</tbody>
</table>
Example A: Deleting old records from Log Stream

The example below shows the JCL and control cards necessary to delete records from an MVS Log Stream that are older than the specified date.

```
//EX4805A EXEC PGM=IAMJUTIL
//SYSUDUMP DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//SYSIN DD *
MVSLOG DELETE,DATE=2009365,
   LOGSTREAMNAME=IDP.CICS.LOGSTRM
/*
```

Figure 48-7: Example of Deleting Old Records from MVS Log Stream (EX4805A)

Example B: Deleting all records from an MVS Log Stream

The example below demonstrates how to delete all data from a log stream. This might be helpful for test systems, where the journaled data does not have to be saved, or after offloading the data from the log stream.

```
//EX4805B EXEC PGM=IAMJUTIL
//SYSUDUMP DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//SYSIN DD *
MVSLOG DELETE,ALL,LOGSTREAMNAME=IDP.CICS.LOGSTRM
/*
```

Figure 48-8: Example of Deleting all records in a Log Stream (EX4805B)
The SCAN command can be used to determine the contents of IAM journal data sets. A report is generated which describes all of the jobs and steps that have written out data to the specified input journal data set, including dates and times.

```
SCAN
    [DSN=dsn] [,EMPTYOK]
    [,FROMBLOCKID=nnnnnnnnnn] [,FROMBLOCKNO=nnnn]
    [,FROMDATE=yyyyddd] [,FROMTIME=hhmmsssth]
    [,FROMTTR=ttttttrr] [,JLOG]
    [,JOUID=cccccccc] [,JOBNAME=cccccccc]
    [,JOURNAL=journal.dsn] [,LOGSTREAMNAME=name]
    [,RLSID=cccc] [,ROUTER=cccc]
    [,STEPNAME=cccccccc] [,TARGET=cccc]
    [,TOBLOCKID=nnnnnnnnnnnn] [,TOBLOCKNO=nnnnnn]
    [,TODATE=yyyyddd] [,TOTIME=hhmmsssth]
    [,TRANID=cccc] [,TRANNO=nnnnn]
```

**Figure 48-9: IAMJUTIL SCAN Command Format**

<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DSN=</strong></td>
<td>Optional keyword that specifies IAMJUTIL is to scan from an IAM RLS journal only those journal records for the specified IAM dataset. Default is that the journal records for all of the IAM data sets will be scanned.</td>
</tr>
<tr>
<td><strong>EMPTYOK</strong></td>
<td>Optional keyword that sets the return code to 0 even if no records were read due to an empty journal.</td>
</tr>
<tr>
<td><strong>FROMBLOCKID=</strong></td>
<td>Optional keyword that specifies IAMJUTIL is to scan journal records from the specified MVS system logger dataset starting with the specified block id number. Default is to start with the first record in the MVS system logger.</td>
</tr>
<tr>
<td><strong>FROMBLOCKNO=</strong></td>
<td>Optional keyword that specifies IAMJUTIL is to scan journal records from a sequential IAM journal file starting from the specified relative block number. Default is to start with the first block in the input journal file.</td>
</tr>
<tr>
<td><strong>FROMDATE=</strong></td>
<td>Optional keyword that specifies IAMJUTIL will scan records starting with those that have the specified date. The date is specified as a julian date in the format yyyyddd. Default is that the scan will begin with the first record in the journal file, regardless of the date.</td>
</tr>
<tr>
<td><strong>Operand</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>FROMTIME=</strong></td>
<td>Optional keyword that specifies IAMJUTIL will scan records starting with those that have the specified time. The format of the time, which is based on 24-hour period, is hhmmssth. Default is that the scan will begin with the first record in the journal file, regardless of the time.</td>
</tr>
<tr>
<td><strong>FROMTTR = tttttrr</strong></td>
<td>Optional keyword to indicate that only journal records with a TTR of this value or higher are to be scanned. The TTR value can be obtained from IAMJUTIL SCAN or DUMP output. The input value is hex digits with up to 6 digits for the track, and 2 digits for the record number field.</td>
</tr>
<tr>
<td><strong>JLOG</strong></td>
<td>Optional keyword that specifies IAMJUTIL will produce a printed log of all activity performed by the scan command. Default is no detailed log will be produced.</td>
</tr>
<tr>
<td><strong>JOBID=</strong></td>
<td>Optional keyword that specifies IAMJUTIL will scan records for the JES JOBID (usually JOBnnnnn or Jnnnnnnn) specified. Default is that records for all jobs are eligible to be scanned.</td>
</tr>
<tr>
<td><strong>JOBNAME=</strong></td>
<td>Optional keyword that specifies IAMJUTIL will scan only the journal records that have a matching jobname. If multiple jobs with the same name are encountered, then all of the jobs with a matching name will be scanned, unless restricted by other operands. Default is that journal records for all jobs are eligible to be scanned.</td>
</tr>
<tr>
<td><strong>JOURNAL=dsname</strong></td>
<td>Optional keyword to indicate which dataset name of the journal file is that is to be used during the SCAN request. This is required if the journal is currently active by IAM/RLS or IAM/PLEX and can not be specified in JCL. Doing so would cause the job to wait for the specified journal to be free.</td>
</tr>
<tr>
<td><strong>LOGSTREAMNAME=</strong></td>
<td>Specifies the name of the MVS system logger dataset from which IAMJUTIL will scan records. This keyword is required when processing an IAM RLS system logger journal dataset. Default is that IAMJUTIL assumes the data is from a sequential journal dataset.</td>
</tr>
<tr>
<td><strong>RLSID=</strong></td>
<td>Optional keyword to indicate that only journal records with the specified IAM RLSID should be included in the SCAN.</td>
</tr>
<tr>
<td><strong>ROUTER=cccc</strong></td>
<td>Select journal records based on RLSID of the ROUTER IAM/PLEX region. Default is ROUTER RLSID will not be used during record selection.</td>
</tr>
<tr>
<td><strong>Operand</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>STEPNAME=</td>
<td>Optional keyword that specifies that IAMJUTIL will scan only those records with the specified stepname. Default is that journal records are eligible to be scanned regardless of the stepname.</td>
</tr>
<tr>
<td>TARGET=cccc</td>
<td>Optional keyword to indicate that only journal records with the specified IAM/PLEX TARGET RLSID should be included in the SCAN. Default is TARGET RLSID will not be used during record selection.</td>
</tr>
<tr>
<td>TOBLOCKID=</td>
<td>Optional keyword that can be used when scanning journal data from an MVS system logger dataset, that indicates IAMJUTIL will scan only data up to and including the specified system logger record ID.</td>
</tr>
<tr>
<td>TOBLOCKNO=</td>
<td>Optional keyword when scanning from a sequential IAM journal, that specifies IAMJUTIL is to scan all data up that is contained in the blocks up to and including the specified relative block number.</td>
</tr>
<tr>
<td>TODATE=</td>
<td>Optional keyword that specifies IAMJUTIL is to scan all journal records up to and including the specified date. The date is specified as a julian date in the format yyyyddd.</td>
</tr>
<tr>
<td>TOTIME=</td>
<td>Optional keyword that specifies IAMJUTIL is to scan all journal records up to and including the specified time. The time is specified in the format of hhmmssth.</td>
</tr>
<tr>
<td>TRANID=</td>
<td>Optional keyword that specifies IAMJUTIL will scan only those records from an IAM RLS journal for the specified CICS transaction name. TRANID specifies the 1 to 4 character transaction name.</td>
</tr>
<tr>
<td>TRANNO=</td>
<td>Optional keyword that specifies IAMJUTIL will scan only those records with the specified CICS transaction number. TRANNO specifies a 1 to5 digit transaction number.</td>
</tr>
</tbody>
</table>
Example A: Basic SCAN of a Journal

This example shows how to run a scan of the current journal data set. The report produced will contain a list of all the job/job steps that updated the corresponding IAM data set, along with time stamps of the first and last record for each job step.

```
//EX4806A EXEC PGM=IAMJUTIL
//SYSPRINT DD SYSOUT=*  
//IAMJRNL DD DSN=my.iam.ksd.log,DISP=SHR
//SYSIN DD *
  SCAN
/*
```

Figure 48-10: Example of Obtaining a Summary of Journal Contents (EX4806A)
80 IAM VERSION 9.2 MESSAGES AND CODES

80.01 IAM Version 9.2 Messages and Codes
This section describes the various printer, console, and TSO messages that may be displayed by INNOVATION's IAM product and the various ABEND codes with which it may terminate.

The general format of IAM messages and WTORs is as follows:

- **IAMnnn** Message on printed report.
- **IAMWnn** IAM WTO/R message.
- **IAMRnnnn** Automatic AIX Recatalog process message
- **IAMJnn** IAM Journal Exit WTO message.
- **IALLnnnn** IAM/RLS and IAM/PLEX messages, displayed on the RLSLOGDD, and may also be a WTO message.

**IAM Utility Return Codes**
IAM utility programs pass a return code at the end of the step, unless they ABEND. A return code of zero indicates that IAM has performed all functions successfully. A return code of thirty-two (032) indicates that the testing period for a trial version of the product has expired. The production version of the product library does not contain an expiration date. Any other return code is accompanied by error messages, and indicates that some kind of error has occurred during the execution.

**VSAM Return Codes**
IAM's VSAM Interface (VIF) will pass return codes comparable to the return codes set by VSAM. The return codes are passed in the RPL for I/O requests or in the ACB for Open or Close requests. The return codes that IAM will issue are documented in Section 80.21 and Section 80.22.

**ABEND Codes**
When using IAM through the VSAM interface, IAM abides by general VSAM rules, which is to pass back error return codes and error codes rather than abending. The exception to that rule is a possible U0184 abend that can occur for diagnostic purposes for various errors when the user has added a //IAMDEBUG DD DUMMY card to a potentially failing job step. This DD statement should only be coded at the request of IAM support personnel to obtain additional documentation for a particular problem. This is because many normally occurring errors may result in a U0184 abend when the DD card is present.

IAM abend codes range from U0100 to U0999. IAM ABENDS are preceded by an IAM error message. For the VIF interface, IAM avoids intentionally abending whenever possible, but will issue appropriate error codes.
80.02 IAM WTO Messages

IAMW01 DD='ddname' I/O ERROR CODE=X'decb iobecb csw cccchhhhr'

Reason: This message is provided by the IAM processor for Compatible format files, when a file access fails with an I/O error. The message contains the following diagnostic information:

DECB - displays the two error bytes of the BDAM DECB(+1). Possible codes are:

8000 - Block not found on track
4000 - Block length was incorrect
0800 - Media failure - Data or Equipment check
0400 - Physical End of File marker
0200 - Unidentified error
0010 - Requested block not within file extents

IOBECB- displays the first four bytes of the IOB.

CSW - displays the channel status word.

ccccchhhhr- displays the cylinder, head and record head number the error was detected on. The head and record number may not be accurate. IBM's message IEA000I/IOS000I, if present, contains the real track address.

Return Codes: A Return Code of 12, X'0C', and a reason code of X'04' for a read I/O error, or X'10' for an output I/O error, is stored in the RPL. The SYNAD exit will be invoked, if so specified by the program.

Action: If an IBM IEA000I/IOS000I message appears on the JCL LOG, a hardware error has occurred. The IBM message gives the sense information (ex: data check, equipment check). Examine the error information provided in the IBM message to determine the cause of the error. The format of this information is documented in the IBM data management SRL for the operating system in use. The file in question is unreadable in its present state and must be reestablished. This may be the result of a hardware error and if possible the new file should be allocated to a different physical location or volume.

For other I/O errors, those not accompanied by a hardware failure error message, determine the cause of the error. It is recommended that an IAMRECVR DIAGNOSE function be executed on the problem file, to see if there are any problems with the file integrity. Some common reasons for these types of I/O errors include:

- File has been improperly moved or restored to a device type different than it was originally loaded on. Frequently, this will fail with block not found I/O error, with a DECB error code of x'8000'.
- Multivolume file has been improperly moved, or improperly cataloged. Frequently, this will result in an I/O error of block not within extent, DECB error code of x'0010'.
- Storage overlay of IAM I/O control blocks. When this type of error occurs, most jobs and application programs do successfully process the IAM file, as does IAMRECVR. The failure is typically limited to one or a few jobs. A SYSUDUMP will be needed to determine the cause of this type of error.
The utility IAMRECVR may be used to recover a file that is no longer usable due to I/O or logical error conditions. This utility can be used to off load records from those portions of the file that have not been physically damaged. The user program should make a decision on the action to take when this occurs, for example to continue processing without this file or to terminate processing until this file is made available. The appropriate action will depend entirely on the application and the user program's evaluation of the diagnostic information that is returned. If further assistance is needed, contact Innovation Data Processing.

IAMW02  REPLY RETRY WAIT OR CANCEL FOR ENQ WAIT ON 'dsname'

**Reason:** The IAM file is not available to this job because some other job is currently accessing the file. This WTO will only be issued when the IAM Global Option VSAMTWO=YES is specified, and is only applicable to Compatible Format files. A load process is always protected against concurrent access (read or update) or from another load. The Share Options specified when the file was defined determine which types of access can concurrently share the file.

**Action:** The operator reply determines the course of action IAM will take. The following responses are allowed:

RETRY: IAM retries the ENQUEUE for the dataset. If the file is still unavailable, the message will be re-issued.

WAIT: IAM waits for the dataset to become available. Caution: The job could time out.

CANCEL: IAM fails the OPEN request with a return code indicating the file is not currently available for processing. The ACBERFLG field will be set to x'A8'.

IAMW03  DD='ddname' FILE FULL DUE TO INSUFFICIENT CORE

**Reason:** This message is displayed by IAM when it finds an insert has failed because of insufficient storage to expand the overflow index. The record that was to be added was not placed in the file. Space may still be available at other locations within the file. Consequently, subsequent inserts may or may not fail depending on where in the file they are placed. Additionally, changes in the storage available within a region may allow a subsequent GETMAIN to succeed.

This message is displayed a maximum of 10 times per execution.

**Return Codes:** A return code of eight (8) and a reason code of 28 (x'1C') is stored in the feedback field of the RPL. This error code indicates a VSAM file full logical error. The logical error exit, LERAD, will be invoked, if so specified in the program.

**Action:** This message indicates that there is an insufficient above the line storage region available for IAM to expand the overflow index size. Reorganizing the file may help to reduce the storage required. For Compatible format files, the definition of the file's OVERFLOW parameters as well as the OCOREO% and OCOREX% parameters should be reviewed. Increasing the OCOREO% value to acquire a larger overflow index area when the file is Opened may correct the problem. The job step region size or, for CICS users, the MVS IEFUSI exit, may also need adjustment so more storage will be available for use by this particular task.
IAMW04 DD='ddname' OPEN ERROR - DATASET IN USE BY JOB jobn

Reason: IAM was unable to open the specified file because it was already opened by another job. For Enhanced Format files, if IAM is able to determine which JOB or user has the file, it will be identified in the message.

Return Codes: A reason code of 168(x'A8') is stored in the ACB error flags field (ACBERFLG) and the OPEN is failed with a return code of 8.

Action: Rerun the job when the indicated job has terminated or closed the IAM file that was in use.

IAMW05 DD='ddname' OPEN ERROR - MORE THAN 2 OPENS FOR UPDATE IN TASK

Reason: This message is displayed by IAM, for Compatible format files only, when it finds a task has issued multiple Opens for update against the same dataset.

Return Codes: A reason code of x'F8' is stored in the ACB error flags and the OPEN is failed with a return code of 8.

Action: Convert the file to an Enhanced Format file, which supports multiple ACB's within the same address space. Or, change the program to insure that the same file is never opened for update more than once within the processing task without first closing it.

IAMW06 DD='ddname' OPEN ERROR - DD STATEMENT MISSING

Reason: This message is displayed when a task has issued an OPEN, but the 'ddname' the OPEN was issued against did not exist, or some other error in open processing has occurred. This message may be preceded by another message if it was due to an error encountered in open processing.

Return Codes: Reason code of x'80' is stored in the ACB error field and the open request is failed with a return code of 8.

Action: Check for other messages that may indicate the cause of the problem. Also, check to see if the missing DD statement is not the result of a misspelling. If not, add the DD statement to the JCL.

IAMW07 DD='ddname' I/O ERROR CODE=X'decb iobecb csw cccchhhhhrr'

Reason: This message is displayed for Compatible Format file when a file OPEN fails with an I/O error. The format is the same as the IAMW01 message, please refer to that message for an explanation of the error codes.

Note: This message may also occur when an attempt is made to OPEN a non-IAM or non-VSAM dataset as VSAM.

Return Codes: The ACB error flag is set to a value of x'B8', and the open is failed with a return code of 8.

Action: Refer to message IAMW01 for potential causes and corrective action.
IAM WTO Messages

IAMW08  DD='ddname' SHORT ON STORAGE IN REGION, REQ:

**Reason:** This message is displayed by IAM when a file OPEN fails because storage is not available for data buffers, the index or control information. Typically this message is issued due to insufficient storage above the 16 megabyte line (31-bit addressable storage), although it can also be issued for shortage of storage below the 16 megabyte line. Below the line storage may have become depleted due to insufficient region specified for above the line storage. The amount of storage requested in kilobytes is provided in the message.

The message can also occur if there was insufficient storage available in the Index Space when a z/OS data space is being used, in which case REGION is replaced by DSPACE, or from 64-bit virtual in which case 64BIT is indicated in place of REGION.

**Return Codes:** Reason code of x'88' is stored in the ACB error flags, and the open is failed with a return code of 8.

**Action:** This message can occur for any number of reasons, and be corrected in various ways depending on the base cause of the problem. Frequently the problem is that the number of records in Independent or Extended Overflow result has significantly increased, causing a high demand for virtual storage for the index to the Overflow area. That problem can most easily be resolved by reorganizing the file. Please refer to the Section 10.51 on Storage Tuning for complete information on IAM's storage usage, and how that usage can be controlled.

IAMW09  DD='ddname' OPEN ERROR - FILE ALLOCATED BUT NOT LOADED

**Reason:** This message is displayed by IAM when a task has issued an OPEN against an IAM file but the dataset (referenced by 'ddname') for input or update processing, but the file had never been loaded as an IAM file, or an attempted load failed.

**Return Codes:** Reason code of x'A0' is stored in the ACB error flags, and the open request is failed with a return code of 8.

**Action:** The file must be successfully loaded with data before it can be used for input or update processing.

IAMW10  DD='ddname' OPEN ERROR - NOT AN IAM/VSAM FILE OR CREATE FAILED

**Reason:** This message is displayed by IAM when a task has issued a VSAM OPEN for the dataset (referenced by 'ddname') but the file did not contain a valid IAM file descriptor block. The dataset may be the result of a load that failed, may have a misspelled dataset name, or perhaps the file was damaged.

**Return Codes:** Reason code of x'BC' is stored in the ACB error flags, and the Open is failed with a return code of 8.

**Action:** First, determine if the file being opened is supposed to be an IAM or a VSAM file. If it is supposed to be an IAM file, review the output from the job which created the file for errors.
IAMW11  DD='ddname' DYNCORE DISABLED DUE TO INSUFFICIENT STORAGE

Reason: This message is displayed by IAM when a user requested storage for IAM's Dynamic Table option and the virtual storage is not available. Processing continues as if the request for Dynamic Tabling had not been requested.

Return Codes: No error codes or return codes are set.

Action: The user should increase the region size or decrease the value requested for DYNCORE.

IAMW12  DD='ddname' DATA COMPRESS ERROR xxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxxxxxx

Reason: During a file access of a data compressed IAM file, IAM decompression failed. The first four bytes are the internal RDW, the remaining data is the key.

Return Codes: A logical error code of x'2C' is set in the RPL. If present and active, the program's logical error exit (LERAD) is invoked, otherwise the request is failed with a return code of 8. If the user has placed a //IAMDEBUG DD DUMMY DD card in the JCL, then the task is abended with an abend code of U0185.

Action: Run an IAMRECVR DIAGNOSE function to validate that the file is still valid. If the DIAGNOSE did not find any errors, then the cause is most likely a storage overlay. If the application did not produce a dump, rerun the job with a //IAMDEBUG DD DUMMY DD card to get a dump. If errors were found by DIAGNOSE, then a file recovery is necessary, either with IAMRECVR or other procedures that have been established. Contact Innovation for assistance.

IAMW13  DD='ddname' FILE FULL, REORGANIZATION REQUIRED

Reason: IAM was unable to add a record into a file, or unable to accept a larger size updated record. For Compatible format files, the Independent Overflow area has been filled. For Enhanced format files, IAM was not able to obtain additional DASD space to expand the size of the file. A message indicating some type of X37 abend may precede this message.

Return Codes: A reason code of 28(x'1C') is stored in the RPL error flags field (RPLERRCD) and the PUT is failed with a return code of 8. This VSAM logical error code signifies a file full error condition. If the application has provided a logical error exit (LERAD), the exit will be invoked.

Action: Reorganize the file. It may be necessary to increase the space allocation for the file, and / or to move the file to a different volume(s). For Enhanced Format files, it may be possible to free up some DASD space that is allocated to other datasets, and retry the request without doing a file reorganization.
IAMW15  DD='ddname' REQUEST FAILED - IAM ADDRESS SPACE QUIESCING OR INACTIVE

Reason: IAM was unable to complete an I/O request for the indicated file that was being processed by an IAMRLS or IAMPLEX address space due to the condition indicated by the message.

Return Codes: For the first message text, the I/O request will receive a logical error code of x'B9' decimal 185. For the second message text the I/O request will receive a logical error code of x'BD' decimal 189.

Action: The application should close the affected IAM file. When the IAMRLS or IAMPLEX region has been reactivated or when it is ok for the application to OPEN the file again, it may then do so.

IAMW16  DD='ddname' OPEN ERROR - IAM DOES NOT CURRENTLY SUPPORT LSR

Reason: For Compatible Format files only, this message is displayed by IAM when a task has issued an open for a dataset with the Local Shared Resources processing option (MACRF=LSR) specified in the ACB, and a UPAD exit active in the exit list (EXLIST) specified for the ACB. Typically, this will occur for a Compatible Format file opened by CICS, without indicating that the file is not to be in any LSR pool in the CICS FCT. The OPEN fails.

Return Codes: Reason code of x'E4' is stored in the ACB error flag field and the open is failed with a return code of 8.

Action: Either convert the file to an IAM Enhanced Format file, which will support the LSR application, or if the error occurred under CICS, change the CICS FCT table to specify LSRPOOL=NONE.

IAMW17  DD='ddname' ADD FAILED - MORE THAN 1 ACB CONCURRENTLY UPDATING FILE.

Reason: IAM does not support concurrent file update. This message will be displayed when IAM detects an inconsistency between an Overflow block, and the index entries for that block in virtual storage. Generally such inconsistencies should only occur if another job had updated the dataset since it was last OPENed by the failing application. IAM expects proper procedures to be used to prevent concurrent update as it will eventually result in damage to the file. This message is displayed a maximum of 10 times per execution.

Return Codes: A return code of eight (8) and a reason code of x'1C' (file full) is stored in the feedback field of the RPL. The logical error exit (LERAD) will be invoked, if so specified in the program.

Action: In a multi-processor environment IAM files must be protected, by a global enqueue facility or manual scheduling, from concurrent update by tasks running on different processors. Within a single processor multiple tasks will be prevented from concurrently opening a file for update by the proper use of SHAREOPTIONS. IAM defaults to a cross region SHAREOPTION 1, multiple read access users or one update
user. The use of SHAREOPTIONS 3 and 4 disable IAM's enqueue protection for a file as does the IAMOVRID parameter UPDATENQ=None. Use these facilities with great care. Within a single task for Compatible Format files, use of multiple ACB's against an IAM file will defeat IAM's enqueue protection for that file. To preserve the integrity of your IAM files any application program that accesses an IAM Compatible Format file with multiple ACB's must be changed to ensure only one ACB is used to access the file for update, or convert the file to the IAM Enhanced File format. (For further assistance contact INNOVATION).

IAMW18 DD='ddname' USER NOT AUTHORIZED FOR UPDATE

Reason: For Enhanced Format files, IAM issued a RACROUTE to verify that the user had authority to access the dataset for the desired mode, and received a return code indicating that the user is not authorized for the requested function.

Return Codes: A reason code of x'98' is stored in the ACB error flags field, and the open request is failed with a return code of 8.

Action: Correct the error and rerun the job. Error could be caused by an incorrect dataset name on the DD card, or this is a legitimate error.

IAMW20 DD='ddname' OPEN ERROR -- error description

Reason: The file load process has detected an invalid, unsupported, or inconsistent parameter for the indicated file attribute. The particular attribute that is invalid is indicated in the error description text. The possibilities are:

- LRECL Specification - Must be at least as long as key offset (RKP) plus the key length, and no larger than 32,755.
- Key Length Specification - Must be greater than 1, and less than 250.
- Blocking Factor Specification - Must be between 1 and 15.
- Blocksate Specification - Must exceed the maximum record length by at least five bytes, and be at least 300 bytes.
- RKP Specification - Must be less than 4092 bytes.
- Independent Overflow Specification - For compatible format files, cannot be a value that will cause the Overflow area to exceed 64,000 blocks.
- Integrated Overflow Specification - Must be between 0 and 99.
- Prime Extension Specification - For Compatible Format files, cannot exceed 32,767.
- Delete Processing Request - Should not occur with the IAM / VSAM Interface.
- Data Control Area (DCA) Validation - Should not occur with the IAM / VSAM Interface.
- Block size Calculation - The IAM internal block size calculation resulted in a value that cannot be used on the device that contains the IAM file.
- IAM File cannot be a DFSMS Striped dataset. – IAM files cannot have multiple stripes, and Compatible format files cannot be SMS Extended format.
- DSNTYPE Invalid for Compatible – Compatible format IAM files cannot use Large Format datasets.
• DSNTYPE=LARGE Requires ZOS 1.7 – Large Format datasets require a minimum operating system level of z/OS 1.7.

• ACBSTRNO > 1 – The program specified a string number greater than one on a file load, which is not allowed for problem state programs. A reason code of 160 (x’A0’) is stored in the ACB error flags field.

• Not licensed for RRDS support – May occur during an attempted load an RRDS file when the version of the IAM product being used has not been licensed for the optional RRDS feature.

• RRDS Initialization Failed – Should be accompanied by additional error message, such as an IAMW36. Please refer to other message(s) for additional details.

• LOAD ERROR -- EXCEEDED 255 EXTENTS - The write (or put) request is failed with a Logical Error code of x’1C’, file full logical error, and the request will receive a return code of 8. If the error occurred during CLOSE, then the CLOSE will fail with a return code of 4, and the ACB error flags set to x’90’. The file is left marked in an unloaded state.

Return Codes: Unless otherwise noted above, a reason code of 192(X’C0’) is stored in the ACB error flags field (ACBERFLG) and the open is failed with a return code of 8.

Action: Correct the invalid specification, and rerun. This error may occur when using VIF if the IAM file was allocated through some means other than an IDCAMS DEFINE and the user failed to correctly provide file specifications through a CREATE Override Control Statement. Additionally, conflicts between the file itself and the user program's internal file definition must be resolved.

IAMW21 DD='ddname' REORG RECOMMENDED: > REQUESTED OVERFLOW nnn,nnn,nnn

Reason: A file reorganization is recommended because the number of records in overflow exceeded the amount specified via the Overflow (O=) override when the file was defined or loaded. The current number of records in the overflow area is included in the message.

Return Codes: None.

Action: Presumably, the override was provided to indicate when the file should be reorganized. IAM will continue to add records to extended overflow, provided that IAM is able to obtain sufficient DASD space to do so.

IAMW22 DD='ddname' REORG RECOMMENDED: OVERFLOW INDEX > nnn,nnn MEG

Reason: A file reorganization is being recommended because the estimated amount of storage required for the overflow index has exceeded the value specified in the IAM Global Option for the IAMW22LIM operand, which defaults to 64 megabytes. The current estimated amount of storage in megabytes is provided in the message. The value is an estimate only, the actual amount used may be different.

Return Codes: None.

Action: The performance on the indicated file may be adversely affected due to the size of the overflow area. In particular, it may take several minutes to open the file, and sequential processing may be detrimentally affected. It is therefore recommended that
the file be reorganized at the earliest convenient time to prevent further performance deterioration. The file can be quickly reorganized with FDRREORG®, or if that product is not available, then use IDCAMS REPRO.

**IAMW23**  
**DD='ddname' OPEN ERROR -- FILE WAS RELOADED SINCE READ ONLY ACB OPENED**  
**Reason:** An attempt was made to open a file with a second ACB (and possibly second DD card) within the same address space for UPDATE processing. The file had previously only been eligible for read only processing. On the open for update, it was determined that the file created time stamp is different from when the file had originally been opened by the READ only ACB. This would indicate that the file has been reloaded, and the index structure cannot be updated.

**Return Codes:** A reason code of 240(x'F0') is stored in the ACB error flags field (ACBERFLG) and the open request fails with a return code of 8.

**Action:** Close all other ACB's open to that dataset, and then reopen.

**IAMW24**  
**DD='ddname' OPEN ERROR – AN ABEND OCCURRED DURING OPEN**  
**Reason:** While IAM was opening a file an ABEND occurred. The ABEND is identified by a preceding IAMW73 error message, which contains the PSW and registers. Open processing for this file is terminated.

**Return Codes:** The OPEN fails with a return code of 8 and a reason code of 188(X'BC') in the ACBERFLG field of the ACB being opened.

**Action:** Contact Innovation Data Processing. Please have all of the available error messages available, along with any dumps to diagnose the problem.

**IAMW25**  
**DD='ddname' INVALID COMPRESSION DICTIONARY BLOCK: nnn,nnn,nnn**  
**Reason:** IAM was attempting to open a dataset with an internal customized hardware compression dictionary. IAM encountered an invalid or corrupted block that should have had a portion of the hardware compression dictionary. The invalid block number is displayed in the message.

**Return Codes:** The OPEN fails with a return code of 8 and a reason code of 188(X'BC') in the ACBERFLG of the ACB being opened.

**Action:** Attempt a dataset recovery, by either using IAMRECVR, or restoring the dataset from a backup. Before overwriting the corrupted dataset, back it up and notify Innovation of the error. Also run IAMRECVR with two print statements: a PRINT IDPINQ and a PRINT FBLK=nnnnn,MAXBLKS=1 where the FBLK value is the block identified by the IAMW25 message. Email that output to Innovation technical support.

**IAMW26**  
**DD='ddname' EXTEND CVAF READ (or WRITE) FAILED R15=xx STATUS=ddd**  
**Reason:** While IAM was expanding the size of a DFSMS Extended Format IAM dataset, the attempt to update the LSTAR in the VTOC failed. Refer to DFSMSdfp Diagnosis Reference for information on the return code and status value.
Return Codes: If an IAMDEBUG DD DUMMY dd card is in the affected job step, then the job step will abend with a U0184, otherwise IAM will continue processing the dataset.

Action: Contact Innovation with the error codes. If problem is repeatable, add in an IAMDEBUG DD DUMMY and a SYSABEND dd card to the job to get a dump at the time the error is detected.

IAMW27 DD='ddname' OPEN ERROR: error description

Reason: While attempting to open an SMS Extended Format or a Sequential Large Format IAM dataset, IAM encountered the error as described in the text of the message. This message will be followed by a IAMW06 message.

Return Codes: Reason code of x'80' is stored in the ACB error field and the open request is failed with a return code of 8.

Action: If unable to correct the indicate problem, rerun the job with an //IAMDEBUG DD DUMMY card and an //SYSABEND DD SYSOUT='' card in the failing job step to produce a dump, and contact Innovation for assistance.

IAMW28 DD='ddname' INELIGIBLE FOR BACKUPCOMPRESSED, OPTION IGNORED' VERIFY FAILED, CONTINUING DUE TO NOVERIFY OVERRIDE IARV64 RC=xxxxxxxx REAS=xxxxxxxx OPEN CONTINES UNABLE TO OBTAIN 64 BIT STORAGE, CACHE64 IGNORED RACROUTE RC4 NO SECURITY DECISION, OPEN CONTINES

Reason: This is a warning message indicating that an error was detected during open processing and was resolved as indicated. Some of these messages are the result to appropriate conditions not being available for an IAM Global Option or Override specification to be used. The RACROUTE RC4 message comes out when the SAFRC4 IAM Global Option is enabled, and indicates that the RACF was not available, and no other security software handled the request.

Action: IAM continues file open processing without use of the indicated function. For the RACROUTE RC4 message, if you are using ACF/2 for security, see the ACF Security topic in Section 90.20 of the IAM manual to make sure that it is properly set up for IAM.

IAMW30 DD='ddname' OPEN ERROR -- FILE IN USE BY JOB jobname

Reason: While attempting to load an IAM file, it was determined that the IAM file is currently in use by some other job or an attempt was made to simultaneously load multiple IAM files to the same dataset. IAM attempts to determine the name of the job using the dataset, and will display the first job found that has the dataset open in this message. If the owning job is not found, then *UNKNOWN is displayed.

Return Codes: A reason code of 192 (x'C0') is stored in the ACB error flags field (ACBERFLG) and the open request is failed with a return code of 8.

Action: An IAM load must be executed without any other job accessing the file at the same time. Wait until the file is closed by the other users, then rerun the file load.
IAMW32  DD=ddname OPEN ERROR – JOURNALING INITIALIZATION FAILED

**Reason:** An error has occurred while attempting to open an IAM file that requested the use of the IAM Journaling exit. IAM was unable to complete initialization for journal processing. This message will be preceded by IAMJxx messages indicating the reason for the failure.

**Return Codes:** A reason code of 188 (x'BC') is stored in the ACB error flags field (ACBERFLG) and the open request is failed with a return code of 8.

**Action:** Review the accompanying IAMJxx messages for the corrective action that is required.

IAMW33  DD=ddname EMPTY ALTERNATE INDEX, DSN=alternate index name

**Reason:** This error message occurs when an alternate index is being opened as an object of a PATH, or as part of an upgrade set, and OPEN determined that the specified alternate index had not been loaded. Valid loading of an alternate index is generally accomplished via an IDCAMS BLDINDEX function.

**Return Codes:** If the alternate index is the object of a PATH that is being opened, the open will fail with a return code of 8, and an error code of 196 (x'C4') in the ACBERFLG field. If the alternate index is being opened as part of an upgrade set, then the open receives a warning return code of 4, and an error code of 100 (x'64') in the ACBERFLG field.

**Action:** Load the identified alternate index datasets with an IDCAMS BLDINDEX, or other program with a comparable function. If the alternate index is no longer needed, make sure that it has been properly deleted.

IAMW34  DD=ddname OPEN ERROR - PLEX/RLS NOT ACTIVE RLSID=rlsid
OPEN ERROR - PLEX/RLS PC ABENDED RLSID=rlsid
OPEN ERROR - PLEX/RLS OPEN FAILED RLSID=rlsid
OPEN ERROR - CICS RLS EXIT NOT INSTALLED
OPEN ERROR - PLEX/RLS REQUIRED: RLSID=rlsid

**Reason:** This error message may occur when attempting to open a file to an IAM/RLS or IAM/PLEX address space, and it was either not active, or it abended while processing this open request. If it was not active, or if the job had previously connected to a different IAMRLS address space then the four character identifier of the IAM RLS address space that was required is included in the message.

**Return Codes:** The open will fail with a return code of 8. If the IAMRLS address space is not active, the ACBERFLG field will be set to 182(x'B6'). If the IAMRLS address space abended, then the ACBERFLG field will be set to 167(x'A7').

**Action:** If IAMRLS is not active, then start it. If it abended, review the log and any dump(s) that may have been taken. Contact Innovation if assistance is required.

IAMW36  DD=ddname TCBTOKEN JSTCB FAILED, RC=xx
DSPSERVE CREATE FAILED, RC=xx REAS=xxxxxxxx
ALESERV ADD FAILED, RC=xx
EXCEEDED MAXIMUM OF 16 DATA SPACES  
RRN BITMAP OVERFLOWED WORK FILE DATASPACE

Reason: One of the above errors occurred while loading an IAM RRDS dataset. The requestor will be failed with an error code indicating insufficient virtual storage.

Return Codes: For a PUT or WRITE I/O request, it will fail with a return code of 8, and a logical error code of 40(x'28'). If the error occurred during OPEN processing, the OPEN will be failed with a return code of 8 and an error code in ACBERFLG of 192(x'C0').

Action: Contact Innovation technical support for assistance.

IAMW37  DD=ddname I/O ERROR ecbx,cswxxxxx,ccw,rd,rbnxxxx,sense ]

Reason: This message is displayed when an I/O error has occurred processing an Enhanced Format IAM file, or during a load of an IAM file. This message may also be accompanied by an IEA000I/IOS000I error message. The message contains the following information separated by commas:

ECB - displays the first 2 bytes of the ECB as posted. Possible values of the first byte include:

- 41 - I/O terminated with error. CSW and / or Sense bytes are useful.
- 42 - Block is not within DASD extents for dataset.
- 47 – DFSMS Extended Format block-id mismatch
- 4E – Attempted write of a block of all hex zeros.

The second byte may contain additional status information relevant for z/HPF architecture channel programs.

CSW - contains the 4 bytes in hex of the status bytes from the CSW. Possible values of the first two bytes for CCW type of I/O are:

- 0C00 - Normal Status
- 0E40 - Unit Check and/or Wrong length record
- 0D40 - End of file

The type of channel program, either CCW (standard I/O) or TCW for z/HPF I/O Indicates either RD for read I/O, or WR for write I/O

The relative block number if known, in hex

SENSE - contains the first two sense bytes from the IAM internal IOB, or if extended sense byte data is present will contain total of 32 bytes continued on the next line. Possible values include of the first 2 bytes are:

- 8000 - Command reject, the device or control unit did not recognize the command
- 4000 - Device requires operator Intervention
• 1000 - Equipment check
• 0800 - Data check
• 0040 - Invalid Track Format
• 0020 - End of cylinder
• 0008 - Block not found

For file loads, the job step will be abended with a U0233 abend code.

**Action:** If an IBM IEA000I/IOS000I message appears on the JCL LOG, a hardware error has occurred. The IBM message gives the sense information (ex: data check, equipment check). Examine the error information provided in the IBM message to determine the cause of the error. The format of this information is documented in the IBM data management SRL for the operating system in use. The file in question is unreadable in its present state and must be reestablished. This may be the result of a hardware error and if possible the new file should be allocated to a different physical location or volume.

For other I/O errors, those not accompanied by a hardware failure error message, determine the cause of the error. It is recommended that an IAMRECVR DIAGNOSE function be executed on the problem file, to see if there are any problems with the file integrity. Some common reasons for these types of I/O errors include:

• File has been improperly moved or restored to a device type different than it was originally loaded on. Frequently, this will fail with block not found I/O error, with a SENSE error code of x'0008'.
• Multivolume file has been improperly moved, or improperly cataloged. Frequently, this will result in an I/O error of block not within extent, ECB error code of x'42'.
• Storage overlay of IAM I/O control blocks. When this type of error occurs, most jobs and application programs do successfully process the IAM file, as does IAMRECVR. The failure is typically limited to one or a few jobs. A SYSUDUMP will be needed to determine the cause of this type of error.

The utility IAMRECVR may be used to recover a file that is no longer usable due to I/O or logical error conditions. This utility can be used to off load records from those portions of the file that have not been physically damaged. The user program should make a decision on the action to take when this occurs, for example to continue processing without this file or to terminate processing until this file is made available. The appropriate action will depend entirely on the application and the user program's evaluation of the diagnostic information that is returned. If further assistance is needed, contact Innovation Data Processing.

**IAMW38**

**DD='ddname' DSPSERV CREATE FAILED, RC=xx REAS=nnnnnnnn

**Reason:** IAM attempted to create a data space for holding the index structure during the file load, however the request was rejected by MVS for the indicated return code and reason code. This is an informational message only, IAM will attempt to continue processing, and utilize a dynamically allocated temporary work file on DASD. For information on the return code and reason code, review in the IBM MVS/ESA Authorize Assembler Services Reference Manual, under the DSPSERV macro.

**Return Codes:** No error codes are set for this situation.
**Action:** Contact INNOVATION for assistance to resolve the problem if unable to do so after reviewing the return code and reason code provided. One of the common reasons for this error message is that the installation exit has either disallowed the use of data spaces, or limited the size. If the size is limited, reduce the IAM Global Option value for DATASPACE, or set it to 0 to prevent the use of data spaces.

**IAMW39 DD='ddname' ###### RECORDS ACCEPTED PRIOR TO ABEND**

**Reason:** IAM has determined that the task loading the specified file has abended, or hit some other type of error condition, after loading the indicated number of records. This information may be useful for determining how much to adjust the space parameters if some type of Sx37 abend has occurred. The number of records that were actually written to the file may be slightly less than indicated due to buffering. *Please note that the IAM file MUST BE RELOADED SUCCESSFULLY before attempting to otherwise access the file.*

**Action:** Correct the error condition as indicated by the abend, and rerun the job. Depending on the cause of the abend, the file may have to be deleted and redefined, for example if a larger space requirement is necessary.

**IAMW40 CARD IMAGE --*cc......cc*  

**Reason:** The input control statement(s) read from the ‘IAMOVRID’ DD statement is displayed when the control statement(s) contains an error or if requested by the user via the 'LOG=YES' operand.

**Action:** If any error condition was raised, another message will indicate the reason for the error. Otherwise, no action is necessary.

**IAMW41 CONTROL STATEMENT OPEN FAILED -- DDNAME='ddname'**

**Reason:** The DDNAME listed was required as control statement input to the override processor, IAMOVRID. An OPEN was attempted, but failed. Processing of the override service is terminated. Normal processing continues.

**Action:** Review the execution job log messages for more detail on cause of the failure. Correct the ‘ddname’ statement and, if necessary, rerun the job.

**IAMW42 INVALID CONTINUATION CARD**

**Reason:** User coded a delimiting comma following the last keyword on a control statement input to the override processor, IAMOVRID, and neglected to provide the next logical record. Processing of the override service is terminated. Normal processing continues.

**Action:** Correct the control statement. The job will continue to run, but may fail or perform unsatisfactorily if the Override data is critical.

**IAMW43 I/O ERROR READING CONTROL STATEMENTS -- DDNAME='ddname'**

**Reason:** An I/O error occurred reading the dataset referenced by ‘ddname’. IAMOVRID is terminated, but normal processing continues.
Action: Examine any system message(s) to determine the cause of the error. The format of system messages is documented in the IBM MESSAGE SRL for the operating system in use. The job will continue to run, but may fail if the Override data is critical.

IAMW44  CONTROL STATEMENT BYPASSED -- 'error description'

Reason: An error was encountered by the override processor, IAMOVRID, during the processing of user supplied Override Control statements. The error description will be from the following list:

- DDNAME NOT SPECIFIED -- The DDNAME operand was missing or misspelled. DDNAME is required to relate the override to a specific IAM file.
- MAXIMUM GLOBAL OVERRIDES EXCEEDED -- The in storage table which holds the Overrides is full. A maximum of 200 control statements may be specified.
- INSUFFICIENT STORAGE -- There was insufficient virtual storage for the IAM override processor to acquire for saving the internal format of the overrides.

Action: Correct the Override statements as follows:

- Add a corrected DDNAME operand to the control statement and, if necessary, rerun the job.
- Reduce the number of global overrides to 200 control statements or less. If more Override statements are required, contact INNOVATION technical support for a modification to IAMOVRID to expand the in-storage table.
- Increase the amount of region, particularly for above the line storage. The override processor needs 48K of storage for the override table. Additional storage will also be required for processing the dataset being opened, so just an increase of 48K will most likely not be sufficient.

IAMW46  'ddname' OPEN FAILURE -- IAMNINFO PROCESSING TERMINATED

Reason: The output report 'ddname' statement could not be opened by IAMNINFO. This DDNAME is usually 'IAMINFO', but may have been overridden by the user. Processing continues without interruption with the IAMINFO report bypassed.

Action: Correct the allocation of DDNAME 'ddname' so the next execution of the job will produce the IAMINFO report.

IAMW47  I/O ERROR MONITORING DSN - 'dsname' - IAM MONITOR TERMINATED

Reason: An IAM monitor facility processor encountered an I/O error while writing monitor/trace data. Monitoring has been discontinued. Normal IAM processing continues.

Action: If a Monitor report is needed, correct the cause of the I/O error and rerun the job.
IAMW48 IAMINFO PARAMETER LIST ABSENT OR IN ERROR -- PROCESSING TERMINATED

**Reason:** The parameter list required by the dynamic file status display processor, IAMNINFO, was missing, was overlaid or is in error. This maybe an internal error. Processing continues without the IAMINFO reported printed.

**Action:** If you are unable to determine the reason for the message, call INNOVATION for further assistance.

IAMW50 IAM VTOC ACCESS FAILED COMP=xxxx CODE=xxxx CLUSTER=clusternamexx

**Reason:** During an IDCAMS DEFINE or RECATALOG of an IAM file, an attempt to access or update a VTOC failed. The completion code and return codes from CVAF are displayed in the message.

**Action:** Refer to the IBM manual ‘Common VTOC Access Facility Diagnosis Reference’ for the meaning of the codes. Correct the problem and re-submit. If unable to correct the problem, contact INNOVATION for assistance.

IAMW51 IAMOVRID CONTROL STATEMENT ERROR, CLUSTER=dsname

**Reason:** During an IDCAMS DEFINE of an IAM file, the IAM Override processor detected an error or invalid IAM override card.

**Return Codes:** The DEFINE is failed with a return code of 140 and a reason code of 36. These codes will appear on the IDC3009I message produced by IDCAMS. The DEFINE is failed on an override error to prevent IAM from defining a file with incorrect attributes.

**Action:** Correct the error on the IAM Overrides, and rerun the dataset. Review Section 30 for proper IAM Override parameters.

IAMW52 IAM SHOWCAT INTERCEPT FAILED

**Reason:** The IAM SHOWCAT intercept function within VIF failed.

**Return Codes:** The calling program will be abended with a U0283 abend code. This error would indicate that something has destroyed the IAM VSAM interface table in virtual storage.

**Action:** Obtain a SYSABEND dump and call INNOVATION for assistance.

IAMW53 RECAT FAILED, DATASET NOT CATALOGED CLUSTER=clusternamexx

**Reason:** An error as indicated by the reason in the message occurred during an IDCAMS DEFINE RECATALOG of an IAM dataset.

**Return Codes:** The RECATALOG request will be failed with return codes from the failing service.
**Action:** Make the corrections to the RECATALOG to correct the identified error, and rerun. If unable to determine why the request failed, contact INNOVATION for assistance.

**IAMW54 PROBABLE IAM FILE HAS NOT BEEN DEFINED, DSN=dsname**

**Reason:** The processing program issued a SHOWCAT catalog request for a non-VSAM file that is cataloged, but does not have the IAM information (as established by DEFINE or file load) was not returned. This message is for diagnostic purposes, and will only be issued when an IAMDEBUG DD DUMMY DD card is in the job step of the program issuing the SHOWCAT macro.

**Return Code:** The SHOWCAT is given a return code of 32 (x'20'), indicating that the file cannot be accessed through IAM or VSAM.

**Action:** This is an unexpected error situation, and should be reported to Innovation Data Processing. Please have a LISTCAT ALL output from IDCAMS available when calling.

**IAMW55 IAM NOT LICENSED FOR AIX OR RRDS SUPPORT CLUSTER=dsname**

**Reason:** An IDCAMS DEFINE was attempted for an alternate index, a path, or a numbered (RRDS) type of dataset. While IAM does support those types of datasets, the version of IAM being used has not been licensed for those optional features.

**Return Codes:** The DEFINE request is failed with a return code of 22 and a reason code of 12. These codes will appear in an IDC3009I message.

**Action:** If you have purchased an IAM license for those features, make sure that you are running with the version of IAM that has been licensed. If you believe that the version you are running should have the optional features license, contact Innovation technical support. If you would like to purchase a license for these features, contact Innovation.

**IAMW56 IAM DEFINE OF NON-SUPPORTED FILE TYPE CLUSTER=dsname**

**Reason:** An IDCAMS DEFINE was issued with an indication that the file should be an IAM file, but the file type cannot be converted to IAM. Examples are VSAM RRDS (Relative Record datasets) and VSAM Linear datasets.

**Return Codes:** The DEFINE is failed with a return code of 22, and a reason code of 8. These codes will appear in an IDC3009I message.

**Action:** Correct the IDCAMS DEFINE to either change the file type to one supported by IAM, or remove the indication that the file is to be an IAM file.

**IAMW57 IAM ALLOCATION FAILED COMP=xxx CODE=xxxx CLUSTER=clusternname**

**Reason:** An error occurred during an IDCAMS DEFINE of an IAM file. The codes displayed correspond to the return code and reason codes of the IDC3009I message from IDCAMS, and there will also be a IDC3009I message on SYSPRINT with the same codes. There may also be additional IDC or IAMW messages.
Return Codes: The DEFINE request is failed with the return code and reason code given in this message. Some of the more common return codes include:

- 8,38 - dataset already cataloged
- 16,0 - SMS failed allocation request, refer to IGD messages
- 22,8 - IAM does not support type of VSAM file requested.
- 42,0 - MVS DADSM failed allocation request
- 56,6 - User not RACF authorized to define the file
- 58,0 - Obtain of VTOC entry failed
- 58,4 - Specified DASD volume(s) not online
- 68,20 - No space on selected volume
- 140,36 – Invalid IAM Override card
- 140,120 – SMS Compressed dataset cannot be used for IAM
- 176,0 - No space in VTOC
- 184,4 - dataset is allocated to another job or user
- 192,0 - Exceeded maximum allowable IAM record size

Action: Refer to Section 80.20 Catalog Return Codes or IDCAMS error message IDC3009I for meaning of the codes. Correct the problem, and resubmit. It may be necessary to issue an IDCAMS DELETE command before attempting to resubmit the DEFINE.

IAMW58 LISTC INTERCEPT FAILED COMP=xxxx CODE=xxxx CLUSTER=UNKNOWN

Reason: A catalog Locate or LISTC request intercepted by IAM failed and received the specified the completion and return codes.

Return Codes: The request is failed with the indicated return code and reason code.

Action: Refer to VSAM errors message IDC3009I for meaning of the codes. Correct the problem and resubmit.

IAMW59 LOCATE FOR AN IAM FILE FAILED CODE=(cc)xxx [DSN=......]

Reason: Locate for an IAM file failed for one of the following reasons:

1. CODE=S1xxx The IAM SHOWCAT intercept issued a locate which failed with return code xxx on the specified dataset. This form of the message will only appear when there is an IAMDEBUG DD DUMMY coded in the failing job step.
2. CODE=S2xxx The IAM SHOWCAT intercept issued a locate which failed with return code xxx. The locate was issued by CI number, so the dataset name is unknown.

Return Code: The SHOWCAT is given a return code of 32 (x'20'), indicating that the file cannot be accessed through IAM or VSAM.

Action: Make sure that the dataset is still properly cataloged. If not, an IDCAMS DEFINE RECATALOG must be done. If further assistance is required, contact INNOVATION.
IAM DYNALLOC FAILED 'description'

**Reason:** During the processing of an IDCAMS DEFINE for an IAM file, IAM's attempted use of Dynamic Allocation failed for the specified reason. This message presents a brief English description of the error code returned by Dynamic Allocation, which is supplied in the IAMW61 error message. Both messages are printed on the system log. There will also be an IDC3009I message on SYSPRINT, with an appropriate error code.

**Return Code:** The DEFINE request is failed, with a return code and reason code that matches the problem description.

**Action:** Correct the error situation, as described with the matching text below, and rerun the DEFINE.

**Text:** DATASET NAME IN USE BY ANOTHER JOB/USER  
**Reason:** The dataset name has been enqueued on by another job/user.  
**Action:** Through whatever software facilities available, determine which job and/or users are enqueued on the dataset, and rerun the DEFINE upon the completion of the other job/user.

**Text:** VOLUME NOT MOUNTED ON SPECIFIED UNIT  
**Reason:** The specified volume was either not mounted, or was mounted but not on the unit specified by the UNIT= keyword on the IAM override control statement for this file. For non-specific volume requests, (i.e., with VOL(ANYVOL) coded), there were no volumes mounted as storage for the unit name specified on the IAM Override Control statement, or SYSDA.  
**Action:** Mount the required volume, or change the volume and/or unit specification.

**Text:** SPECIFIED UNIT NAME IS UNDEFINED  
**Reason:** The unit name specified on the IAM Override Control statement for this file does not exist on the system that the define was attempted.  
**Action:** Correct the unit name specification, or run on the proper operating system.

**Text:** REQUIRED CATALOG NOT MOUNTED  
**Reason:** The catalog required for the definition of the IAM dataset is on a volume that is not currently mounted.  
**Action:** Insure that the volume containing the user catalog is mounted, and rerun the DEFINE.

**Text:** DATASET ALREADY EXISTS  
**Reason:** The dataset being Defined already exists in the catalog, and may or may not be on the volume it is cataloged to.  
**Action:** Make sure that the cluster name is correct and if not correct it. If it is correct, delete the dataset from the catalog (and volume if applicable) and rerun the DEFINE.

**Text:** DUPLICATE DATASET NAME ON VOLUME  
**Reason:** The dataset already exists on the specified volume, and is not cataloged.  
**Action:** Delete the dataset from the volume, and rerun the DEFINE.

**Text:** NO SPACE IN VTOC  
**Reason:** There was no space in the VTOC (Volume Table of Contents) or the VTOC Index for the new dataset on the specified or selected volume.  
**Action:** Either correct the error by increasing the size of the VTOC or VTOC Index on
the volume (this can be done by use of COMPAKTOR), delete unwanted datasets from the volume, or select a different volume.

Text: **VTOC I/O ERROR OR CVAF ERROR**  
*Reason:* An I/O error occurred on the VTOC during file allocation.  
*Action:* Review SYSLOG for other messages indicating a more precise cause of error. Correct the problem and rerun DEFINE.

Text: **REQUESTED SPACE NOT AVAILABLE ON VOLUME**  
*Reason:* The volume specified or selected did not have sufficient space to satisfy the request.  
*Action:* Ensure that the space requested is actually needed, and adjust if possible. (NOTE: IAM files generally require less space than VSAM files.) Otherwise, select a different volume, remove unneeded datasets from the volume, or run COMPAKTOR to consolidate free space.

Text: **USER NOT AUTHORIZED TO ALLOCATE DATASET**  
*Reason:* The job lacks RACF authorization to DEFINE the dataset.  
*Action:* Contact the Security Administrator for assistance.

Text: **INSTALLATION EXIT REJECTED ALLOCATION REQUEST**  
*Reason:* A dynamic allocation exit routine in the system did not allow the allocation request to be processed.  
*Action:* Correct the DEFINE to the installation requirements.

Text: **REQUIRED CATALOG NOT AVAILABLE**  
*Reason:* The user catalog required may have been deleted or disconnected from the system master catalog, or may have been damaged and is being recovered.  
*Action:* Correct the error with the user catalog, and rerun the DEFINE command.

Text: **DUPLICATE DATASET NAME IN CATALOG**  
*Reason:* The dataset name already exists in the catalog, and may or may not exist on disk.  
*Action:* Make sure the cluster name is correctly specified. If it is delete the current entry from catalog (and disk if applicable).

Text: **NO SPACE IN CATALOG**  
*Reason:* Insufficient space in the catalog to contain the record for the new dataset.  
*Action:* Enlarge the catalog, and rerun the DEFINE.

Text: **SMS FAILED REQUEST. REFER TO PRIOR MESSAGE(S)**  
*Reason:* The allocation request was failed by SMS. There should be preceding messages from SMS indicating the reason for the error.  
*Action:* Correct the problem indicated by the SMS error messages, and try request again.

**IAMW61 IAM DYNALLOC FAILED COMP=nnnn CODE=nnnn CLUSTER=clusternname**  
*Reason:* The dynamic allocation requested by IAM to perform the DEFINE operation failed with the printed error codes. This message may be accompanied by an IAMW60 message.

*Return Code:* The DEFINE request is failed, with a return code and reason code that matches the problem description.
**Action:** Refer to message IAMW60, if printed, and/or the IDC3009I error message on SYSPRINT. The error codes from Dynamic Allocation are documented in the MVS/XA and MVS/ESA System Macro and Facilities manual, the MVS Job Management SPL, and under the ISPF tutorial. Correct the error condition as indicated by the error codes, and rerun the DEFINE.

**IAMW62 IAM OPEN FAILED FOR DDNAME=ddname CLUSTER=clustername**

**Reason:** During DEFINE processing of an IAM file, IAM attempted to OPEN the defined file, however the OPEN failed. Additional IBM messages may appear on the system log.

**Return Codes:** The DEFINE request is failed with a return code of 62, reason code of 0.

**Action:** Determine the cause of the OPEN failure, correct the error, and rerun the job. For a new DEFINE (as opposed to RECATALOG), DELETE and redefine the dataset.

Note: At this point, the dataset has been allocated and cataloged, but is not yet usable by IAM.

**IAMW63 IAM I/O ERROR: 'synad message'**

**Reason:** During the processing of a DEFINE command for an IAM file, an I/O error occurred when reading or writing the IAM control information.

**Return Codes:** The DEFINE request is failed with a return code of 62, reason code of 0. If an IAMDEBUG DD DUMMY is specified, then the program will abend with a U0310.

**Action:** Using the standard SYNAD message and other messages that may appear on SYSLOG, determine the cause of I/O error and correct it. If this was not a RECATALOG operation, DELETE and DEFINE the IAM dataset again.

Note: At this point, the dataset has been allocated and cataloged, but is not yet usable by IAM. For RECATALOG operations, the file is either not a previously DEFINED or loaded IAM file, or there is an error with the dataset requiring recovery. The recovery can be done by restoring the dataset from a good copy or possibly by using program 'IAMRECVR'. Use of the recovery program may result in data loss.

**IAMW64 UNEXPECTED END OF FILE READING AN IAM FILE FOR RECATALOG REQUEST -- NOT VALID IAM DATASET**

**Reason:** During the processing of a DEFINE RECATALOG command for an IAM file, an end of file occurred while attempting to read the file characteristics.

**Return Codes:** The DEFINE request is failed with a return code of 86, reason code of 4.

**Action:** The dataset is empty. The recatalog request was not performed. Either the file was never an IAM file, in which case no corrective action is required, or the dataset has been clobbered. To recover the dataset, it can be restored from a good backup, or a recovery attempted with program 'IAMRECVR'. Recover the file, then retry the recatalog processing.

Note: This message will not be issued by IAM Version 8.0, but is retained in the manual for documentation purposes.
IAM WTO Messages

IAMW65  IAMSCRATCH FAILED COMP=nnnn CODE=nnnn VOLSER=vvvvvv
IAM UNCATLG FAILED COMP=nnnn CODE=nnnn CLUSTER=clusternname

Reason: After an error attempting to catalog or initialize an IAM file being defined, an attempt to delete or uncatalog the dataset failed. The codes are returned from SCRATCH or UNCATLG request, which are documented in the SYSTEM DATA ADMINISTRATION manual.

Return Codes: The return code for the DEFINE is based on the original condition that caused the error.

Action: The dataset is still on the specified volume. Refer to an immediately preceding IAMWnn message for the dataset name. The dataset must be manually scratched from the indicated volume and uncataloged prior to attempting to redefine the dataset.

IAMW66  IAM REALLOC FAILED CODE=nnnn INFO=nnnn CLUSTER=clusternname

Reason: After successfully defining an IAM file, IAM had determined that the job step had DD cards which were allocated to the file, but were allocated to the wrong volume. The attempt to reallocate the file with dynamic allocation failed, with the indicated error codes.

Return Codes: The DEFINE completes with a return code of 0.

Action: The IAM file has been successfully defined, but attempts to REPRO into the IAM file within the same step may fail. A subsequent REPRO into the IAM file can be done.

IAMW67  IAM SMS ALLOC FAILED RC=X'xx' REAS=X'xxxxxxxx' CLUSTER=clusternname

Reason: The define of an IAM file failed using DADSM allocation with the specified return code and reason code. Refer to the DADSM Create (ALLOCATE) Function Return Codes section of the IBM MVS/ESA DADSM/CVAF Diagnostic Aids for a description of the error codes.

Return Codes: The DEFINE request is failed with a return code of 62, reason code of 0.

Action: Take the appropriate corrective action based on the error codes indicated, and retry the define request.

IAMW68  IAM UNIT NAME SEARCH FAILED, RC=xx CLUSTER=clusternname

Reason: During the define of a multivolume nonspecific allocation, an IAM call to the MVS Unit Name look up service failed as indicated in the message. The return code, if provided, is documented in the IBM MVS System Modifications Manual.

Return Codes: The DEFINE request fails with a return code of 72, reason 4.

Action: If the condition indicated by the return code cannot be corrected, contact INNOVATION for support. As a circumvention, try a different UNIT override, or switch to specific volume allocation.
IAMW69 IAM xx ELIGIBLE VOLUMES, nn VOLUMES REQUIRED CLUSTER=clustername

Reason: During the define of a multivolume nonspecific allocation, IAM found the indicated number of storage volumes in the specified UNIT name pool, however, more volumes then available were needed to satisfy the allocation request.

Return Codes: The DEFINE request fails with a return code of 72, reason 4.

Action: Change the UNIT override to indicate a UNIT name that has sufficient storage volumes, or reduce the number of volumes requested.

IAMW70 DD='ddname' PUT ERROR -- WORK FILE DATASPACE OVERFLOW

Reason: The size of the data space used to hold the index structure during a file load was insufficient for the file indicated by the ddname.

Return Codes: A reason code of 244('F4') is stored in the RPL error flags field (RPLERRCD) and the request fails with a return code of 8. The logical error exit, LERAD, if specified will be invoked. If the error is detected during close, the program is abended with a U0246 abend.

Action: Increase the size of the data space by using the IAM CREATE override, specifying the DATASPACE keyword. As an alternative, specify a DATASPACE=0, which will force the use of a temporary work file.

IAMW71 TRACE DEACTIVATED - text indicating reason

Reason: The IAM trace facility for Enhanced Format files has detected an error during activation. The possible reasons include

- The DDNAME IAMATR31 is not available
- There is insufficient storage available to obtain the trace work area.

Action: If the IAMATR31 DD was not specified, add it to the job. If storage was not available for the trace work area, specify a larger REGION size, or a REORG of the IAM file may be required.

IAMW72 IAMASY ESPIE RECOVERY ENTERED FOR ABEND S0Cx

Reason: A program check occurred under the IAM IRB while performing asynchronous processing for an I/O request with RPL OPTCD=ASY. Included in the displayed information are the PSW and registers at time of the error.

Return Codes: The specific request causing the error, if it is identified, will receive a return code of 8 with an RPL error code of 240('F0') stored in the RPLERRCD. IAM will internally issue an ENDREQ for that RPL to clean up any resources it may have held.

Action: IAM will attempt to continue processing for the affected file. Contact Innovation Technical Support with the full text of the message for assistance, and save any dumps that may have occurred to aid in diagnosis.

Note: This message will not occur in IAM Version 8.0. It is presented here for documentation compatibility with prior versions.
IAMW73 IAM module name ESTAE ENTERED FOR ABEND Sxxx

**Reason:** A abend occurred while opening or processing an IAM file. If the error occurred during open processing, IAM will attempt to free the resources acquired during the open process. Depending on the circumstance, the PSW and registers at time of the error may also be displayed.

**Return Codes:** The job may abend with a U0184, or the Open, Close, or I/O request will fail. An open request will fail with a return code of 8, and the ACB that has caused the error will have a reason code of 188('X'BC') set in the ACBERFLG field. If IAM was using a Data Space for this file, it will be included in the dump. An I/O request will fail with a logical error, return code of 8, with an error code of 'x'B8'.

**Action:** Contact Innovation Technical Support with the full text of the message for assistance, and save any dumps that may have occurred to aid in diagnosis. For CICS, IAM will terminate open processing for this ACB, and will attempt to free all resources used by the failing OPEN request.

IAMW74 UCBLOOK MACRO FAILED R15=xxxx R0=xxxx CLUSTER=clusternam

**Reason:** During IAM file definition, an attempt to find the UCB on which the dataset resides, or will reside, using the IBM UCBLook service has failed. IAM will continue, if possible, with the define request by utilizing a different UCB lookup technique.

**Action:** Contact Innovation Technical Support with the full text of the message for assistance.

IAMW77 DD='ddname' OPEN ERROR - DYNALLOC FAILED - CODE=xxxxxxxx

**Reason:** During open of an IAM dataset requiring an open of an alternate index component, the associated data set could not be allocated. The open process will fail with a return code of 8.

**Action:** Look up the dynamic allocation failure code in the IBM Manual on z/OS Authorized Assembler Services Guide and take appropriate corrective action. Refer to Section 10.65 on Special Considerations for IAM Alternate Indexes for more information. If additional assistance is needed, contact Innovation Technical Support.

IAMW78 DD='ddname' OPEN ERROR - AN ABEND CONDITION OCCURRED

**Reason:** While IAM was opening a file to be loaded, some type of system Abend occurred. There should be additional messages indicating the exact abend condition. Open processing for this file is terminated.

**Return Codes:** The OPEN fails with a return code of 8 and a reason code of 192('X'C0') in the ACBERFLG field of the ACB being opened.

**Action:** Correct the abend condition, and then reload the file. The most typical abend condition is an X37 abend, in which case delete and redefine the dataset with more DASD space.
IAMW79   DD='ddname' OPEN ERROR - text indicating reason

**Reason:** IAM was unable to open the specified file due to damaged control or index information on the file.

**Return Codes:** A reason code of 188(X'BC') is stored in the ACB error flags field (ACBERFLG) and the OPEN is failed with a return code of 8. If an IAMDEBUG DD DUMMY is in the job step, then the job will be abended with a U0184.

**Action:** **DO NOT ATTEMPT TO REORGANIZE THE DATASET WITH IDCAMS OR OTHER SOFTWARE!!!** Use the IAMRECVR to recreate the file. For diagnosis, please do the following:

1. Rerun job with an //IAMDEBUG DD DUMMY card added and a //SYSUDUMP or //SYSABEND. If you have the ABENDAID, please insure that the standard IBM dump will be taken.
2. Save the damaged file, or back it up using FDR/ABR, or DFDSS.
3. Contact Innovation for assistance.

IAMW80   MODULE 'modname' INSTALLED AT 'address' - VER nn

**Reason:** The message is issued in response to a VIF status request when the module 'modname' is in place and is ready to provide IAM services to programs using ACB's to access VSAM files. The 'address' given is the virtual storage location of the named IAM VIF module. The version level number is also given for the module.

**Action:** None, information message only.

IAMW81   THE IAM SYSTEM MODULES ARE 'status'

**Reason:** This message is provided by IAMSTART when the IAM VIF modules are installed, or in response to VIF action commands when the IAM system level VSAM interface is already in place. The 'status' of the modules may be:

- ACTIVE
- REACTIVATED
- ALREADY INSTALLED

This message includes the version and release level of the VIF modules installed.

**Action:** None, information message only.

IAMW82   THE IAM SYSTEM MODULES ARE 'status'

**Reason:** This message is provided by IAMSTART in response to VIF action commands when the IAM system level VSAM interface is already in place. The status of the modules maybe:

- NOT ACTIVE (the IAM system level VSAM interface is not in place),
- INACTIVE (the IAM system level VSAM interface is in place but is not active).

**Action:** None, information message only.
IAMW84 IAMSTART ESTAE RECOVERY ENTERED FOR ABEND Sxxx Uxxxx AT OFFSET xxxx

Reason: This message is provided by IAMSTART when VIF ABENDS. An attempt is made to provide diagnostic information.

Action: If the problem persists call INNOVATION for assistance.

IAMW85 'ADDRESS' 'HEX core print'----- 'EBCDIC core print'

Reason: This message is provided by IAMSTART when the VIF modules are first installed and is also the response to a VIF STATUS request when VIF is in place and is ready to provide IAM services to programs using ACB's. The 'ADDRESS' given is the virtual storage location of the VIF Vector Table Entry. The remainder of the line is the entry in hex and display format.

Action: None, information message only.

IAMW86 IDPSTART FAILURE - REASON='number' 'reason'

Reason: The activation of the IAM VSAM Interface failed for one of the following reasons:

- 01 OPERATING SYSTEM NOT MVS OR SP 1.2 OR HIGHER
- 02 INVALID INPUT PARAMETERS
- 03 UNABLE TO OBTAIN AUTHORIZATION
- 04 'vector table name' VECTOR TABLE IS INVALID
- 05 SYSLIB DD MISSING OR OPEN ERROR
- 06 'modname' NOT FOUND IN SYSLIB
- 07 'modname' MODULE LOAD ERROR
- 08 ERROR MODIFYING THE SVC TABLE
- 09 'modname' NOT FOUND
- 10 'modname' NOT FOUND IN THE LINKLIST
- 11 'modname' UNABLE TO DE-INSTALL
- 12 'modname' CDE OR LPDE ABOVE 16M ERROR
- 13 'modname' MODULE ABOVE 16M ERROR
- 14 GETMAIN ERROR
- 15 'modname' HAS SMP INSTALLED IDP MODULE
- 16 UNABLE TO OBTAIN DISPATCHER LOCK
- 17 FREEMAIN ERROR
- 18 UNABLE TO OBTAIN LOCAL LOCK
- 19 'modname' CDE NOT FOUND
- 20 ENQUEUE/DEQUEUE ERROR ON IAM RESOURCE
- 21 ENQUEUE/DEQUEUE ERROR ON SYSZSVC
- 22 INVALID VECTOR TABLE STATUS
- 23 INVALID VECTOR TABLE CDE ADDRESS
- 24 RC=xxxx FROM SVCUPDTE
- 27 'modname' and IAMVECTB ARE OUT OF SYNCH
- 28 VECTOR TABLE CANNOT BE STOPPED
- 31 HIGHER VERSION OF VIF ALREADY STARTED

**Action:** If the problem persists call INNOVATION for assistance.

**IAMW89**

IAM - TRIAL VERSION FROM INNOVATION DATA PROCESSING EXPIRES IN 'nnn' DAYS (PLEASE CONTACT INNOVATION)

**Reason:** This is a trial version of the IAM system. The number of days the trial will remain active is displayed.

Note: This message will never appear if you are a licensed user of IAM. If you are a licensed IAM user, then it is possible that you have in your job a bad STEPLIB pointing to the old trial library. The library that contains the production copy will appear in the heading with a P following the version identification. For example, IAM V9.2/01P.

**Action:** When there are 10 or fewer days before the trial is due to expire this message will become nondeletable. To prevent the trial from expiring call INNOVATION for an extension PARM=value and use the JCL shown below to extend your trial's expiration date. The JCL to extend the trial is as follows:

```
//EXTEND EXEC PGM=IAMEXTND,PARM=xaxx
//STEPLIB DD DISP=SHR,DSN=your.user.lib
//SYSLIB DD DISP=SHR,DSN=your.user.lib
//@BINDNOT DD DUMMY
//SYSDIAG DD SYSOUT=*  
```

**IAMW90**

IOSCAPU MACRO FAILED COMP=xxxx CODE=yyyy

**Reason:** IAM attempted to capture a UCB to initialize the IAM file being defined. The capture of the UCB failed, as indicated in the error message.

**Action:** IAM terminates processing for the file being defined. The file will need to be deleted and redefined before it can be processed. The error codes are available in the MVS Authorized Assembler Services Reference manual. If assistance is needed with diagnosing the problem, contact Innovation Data Processing. Attempting to define the file on a different volume(s) may circumvent the problem. A dump can be obtained by including an IAMDEBUG DD DUMMY statement in the JCL, which will result in a U0310 abend.
IAMW91  DD='ddname' REORG RECOMMENDED: > 13 EXTENTS ON ONLY VOLUME

**Reason:** A file reorganization is being recommended because the dataset has 14 or more extents on a single volume and is limited to 16. It is getting close to receiving a possible file full error.

**Return Codes:** None.

**Action:** Because IAM files have a non-VSAM file structure, they are limited to 16 extents per volume. The file indicated currently has 14 or more extents, so future growth will be restricted. To prevent an out of space condition, action should be taken at the earliest possible time. If there is sufficient space for the file to expand on the volume that it currently resides, either use COMPAKTOR to merge extents, or reorganize the file doing a DELETE and DEFINE of the dataset, specifying a larger space allocation and DSNTYPE=LARGE so more tracks can be used on the volume. The current space allocation values can be determined by performing a Listcat All on the dataset. If the current volume has insufficient space, and you are unable to free up sufficient space, then the dataset should be moved to a different volume, where more DASD space is available.

IAMW92  DD='ddname' REORG RECOMMENDED: OVERFLOW > 1000 CYLINDERS

**Reason:** A file reorganization is recommended due to the amount of space currently utilized for the overflow area. The large amount of space used for the overflow records could impact performance, in particular increasing the amount of time to open the file, and could impact sequential processing times.

**Return Codes:** None.

**Action:** Reorganize the dataset as soon as possible.

IAMW99  INTERNAL LOGIC ERROR -- JOB TERMINATED

**Reason:** IAMOVRID has encountered an illogical condition.

**Action:** Obtain a SYSUDUMP or SYSABEND dump, and call INNOVATION for assistance.
<table>
<thead>
<tr>
<th>IAMR000</th>
<th>IAMRECAT UNABLE TO OBTAIN STORAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reason:</strong> The automatic recatalog processor was unable to obtain storage.</td>
<td></td>
</tr>
<tr>
<td><strong>Action:</strong> Rerun job with a bigger region.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IAMR0002</th>
<th>IAMRECAT GETMAIN FOR COMPAREA FAILED</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reason:</strong> The automatic recatalog processor was unable to obtain storage.</td>
<td></td>
</tr>
<tr>
<td><strong>Action:</strong> Rerun job with a bigger region.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IAMR0003</th>
<th>IAMRECAT BYTE STRING BUILD FAILED</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reason:</strong> The automatic recatalog processor was unable to build the byte string needed to determine the component name changes.</td>
<td></td>
</tr>
<tr>
<td><strong>Action:</strong> Contact Innovation support and provide a listcat all of the base cluster and components.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IAMR0004</th>
<th>IAMRECAT UNABLE TO LOCATE DSN - 'dsname'</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reason:</strong> The catalog locate failed for one of the components the AIX automatic recatalog processor determined should be part of an AIX sphere.</td>
<td></td>
</tr>
<tr>
<td><strong>Action:</strong> Check to make sure that all of the files in an AIX sphere have been renamed and are cataloged.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IAMR0005</th>
<th>IAMRECAT UNABLE TO VALIDATE DSN - 'dsname'</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reason:</strong> The new name of a component generated by the AIX automatic recatalog processor does not match what is in the AIX information area.</td>
<td></td>
</tr>
<tr>
<td><strong>Action:</strong> Check to make sure that all of the files in an AIX sphere have been renamed with a consistent change pattern and are cataloged.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IAMR0006</th>
<th>IAMRECAT INVALID AIXAREA DSN - 'dsname'</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reason:</strong> There is an inconsistency in the AIX info area of one of the components located during the recatalog process.</td>
<td></td>
</tr>
<tr>
<td><strong>Action:</strong> Check to make sure that all of the files in an AIX sphere have been renamed with a consistent change pattern and are cataloged.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IAMR0007</th>
<th>IAMRECAT COMPAREA CHAIN POINTER MISSING</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reason:</strong> An internal pointer in a table created by the recatalog processor is missing or invalid.</td>
<td></td>
</tr>
</tbody>
</table>
**Action:** Contact Innovation support and provide a listcat all of the base cluster and components.

IAMR0008 IAMRECAT RECATALOG OF DSN - 'dsname'

**Reason:** The recatalog of one of the components of an AIX sphere failed.

**Action:** Check to make sure that all of the files in an AIX sphere have been renamed with a consistent change pattern and are cataloged.

IAMR0009 IAMRECAT DSN PARSING FAILED

**Reason:** The automatic recatalog processor was unable to parse the incoming dataset name

**Action:** This should never happen and indicates an error in the name of the dataset being processed.

IAMR0010 IAMAIX AREA HAS NO ENTRIES

**Reason:** The automatic recatalog processor found an AIX component with an invalid AIX info area.

**Action:** This should never happen and indicates an error in the internal AIX information kept in the dataset.
80.03 IAM SYSPRINT Messages

The following general messages are issued by various IAM utility programs. These messages are typically written to SYSPRINT by the originating utility program. These are messages from the IAMRECVR, IAMSIMVS, IAMSMFVS, IAMSMF, and IAMZAPOP utility programs provided with the IAM product.

IAM099 NEAR RELATIVE DATA LOCATION nnnn 'error description'

**Reason:** The common parsing routine encountered an error in parsing the user specified control statements. The approximate location of the error was position nnnn, counting the first position as 000.

**Action:** Correct the error and resubmit job.

IAM100 IAM FILE ANALYSIS - DSN= dataset name

**Reason:** This report is produced from a LISTCAT request on the IAMPRINT DD output, and also on SYSPRINT when IAMRECVR is run against an IAM file. The report contains descriptive information and statistics about the IAM file. For a complete description of the output report following this message, please refer to the LISTCAT Report description in Section 10.71 in the IAM manual.

**Action:** None, this is an informational message only.

IAM213 'file type' RECOVERY FILE CREATED DDNAME='ddname'

**Reason:** IAMRECVR has completed the output of a file to the DDNAME 'ddname'. The 'file type' is as follows:
IAM - An IAM file created by the RECOVER operation.
DUPLICATE - A key/log file containing duplicate records found in the IAM file that is being recovered by IAMRECVR.

**Action:** Refer to the documentation of IAMRECVR for the uses of the different output files it creates.

IAM260 LISTC PROCESSING FAILED FOR DSN='dsname'

**Reason:** An error occurred during IAM processing of an IDCAMS LISTC command for a possible IAM file. This message will be preceded by one or more error messages describing the error in more detail.

**Action:** Review other messages for source of error, and correct as necessary. IDCAMS LISTC processing will continue normally, however no information will appear in the IAMPRINT file for the named file.

IAM262 DATASET IS NOT AN IAM FILE

**Reason:** During an IDCAMS LISTC IAM attempted to process a dataset as an IAM file, however it was determined that the dataset was not an IAM file. The dataset name is displayed in the IAM260 message.

**Action:** None, information message only.
IAM266  LOAD OF MODULE 'modulename' FAILED

**Reason:** During an IDCAMS LISTC command, IAM encountered an error loading the named support module. There should also be an accompanying message in SYSLOG indicating the cause of the problem (i.e. S106, S306, or S806 error). The dataset name is presented in the accompanying IAM260 message.

**Action:** Find the IBM error message, and take corrective action. Possibilities are insufficient virtual storage to load the module, or IAM is not in the system LINKLIST and no STEPLIB has been provided.

IAM269  IAM CPL PROCESSING FAILED

**Reason:** IAM was attempting to process a CATALOG parameter list to determine dataset name and volume information, however the CPL did not contain the expected information. There will be no information listed about the IAM file.

**Action:** Contact INNOVATION for assistance.

IAM303  CARD IMAGE -- * control statement image *

**Reason:** A display of the SYSIN dataset input control statements.

**Action:** None. Information message only.

IAM316  RECOVERY CAN BE FORCED BY SPECIFYING VALUES FOR THE FOLLOWING -

**Reason:** An attempt was made to recover an IAM file that has damaged control records. The unreadable control records requires the user to specify key data normally extracted from the file.

**Action:** The next line(s) displayed will detail the fields required. Use this information to continue the recovery.

IAM318  * WARNING* DATASET-'dsn' DEFINED FIXED CONTAINS VARIABLE LEN RECORDS

**Reason:** The DEFINE for this cluster shows the average and maximum record lengths to be equal. IAMSIMVS detected one or more records that were not equal to the average record length Defined for this cluster.

**Action:** IAMSIMVS continues processing as if the file contained fixed length records. Blocking, overflow and other values established for a file however vary depending upon whether record lengths are fixed or variable. To obtain a more accurate estimate of IAM's space savings for this file, include the 'VARIABLE' keyword with the 'SELECT' option.

**Note:** To properly identify this file as containing variable length records the DEFINE for this file should be changed so the average RECORDSIZE value is less than the maximum. During processing, if the Defined average RECORDSIZE value is not changed, IAM will return a record length error for this file.
IAM319  PREMATURE END OF FILE -- AFTER BLOCK nnnnnnnn
Reason: IAMRECVR detected a premature end of file at block nnnnnnn. Data blocks may have been lost.
Action: Review the contents of the recovered file. A section of the file being recovered may not have been readable.

IAM320  MAXIMUM BLOCKS LOST DUE TO END OF FILE -- nnnnnnnn
Reason: An end of file error erases the remainder of the track. The reported number of blocks could have existed on the track but were not yet read when the end of file was encountered.
Action: Review the contents of the recovered file. A section of the file being recovered may not have been readable.

IAM321  INVALID VARIABLE LENGTH FIELD -- BLOCK nnnnnnnn
Reason: IAMRECVR detected an invalid RDW length field for a given record (i.e.: RDW exceeds maximum LRECL). The block number is printed and the remainder of block is bypassed.
Action: Review the contents of the recovered file. A section of the file being recovered may not have been readable.

IAM322  DROPPED DUE TO I/O ERROR -- BLOCK nnnnnnnn
Reason: IAMRECVR has dropped block nnnnnnnn. The first 24 bytes of the block at the location of the errors is printed in hexadecimal. Processing continues.
Action: Review the contents of the recovered file. A section of the file being recovered may not have been readable.

IAM323  SEQUENCE CHECK -- BLOCK nnnnnnnn
Reason: An out of sequence record was encountered. The IAM block number is printed along with 24 bytes of the key in hexadecimal. Processing continues.
Action: Review the contents of the recovered file. A section of the file being recovered may not have been readable.

IAM324  UNABLE TO CALCULATE LOST BLOCKS - ASSUMING nn MISSING
Reason: An end of file erases the remainder of the track. Any blocks that were on the track at the time the EOF record was created have been destroyed.
Action: Review the contents of the recovered file. A section of the file being recovered may not have been readable.

IAM325  FILE NOT COMPRESSABLE - DATA LEN PAST KEY < 11 BYTES - DSN=
Reason: IAMSIMVS determined that the records contained within the file were not eligible for IAM record compression. IAM only compresses the data in a record located
past the end of the key and only if the length of that data is more than 10 bytes. (i.e. MAXLRECL-(RKP+KEYLEN) > 10).

**Action:** No action is required. IAMSIMVS will continue the simulation for the file as if it were not converted to an IAM file in compressed format.

### IAM325 ERROR DECOMPRESSING RECORD – BLOCK nnnn

**Reason:** IAMRECVR was unable to decompress a compressed record from an IAM dataset.

**Action:** Most probable cause is that a corrupted buffer was written out to the dataset. If possible, recover the file through other means. If that is not possible then continue IAMRECVR will recover all the records that it is able to read, but will drop the rest of the records. Backup the dataset as described in the User's Guide section of the IAM Manual, and contact Innovation for support.

### IAM326 IAMCRTSM ERROR -- 'error description' -- DSN=dsn

**Reason:** IAMCRTSM was invoked to simulate the conversion of a VSAM file and encountered a logical error. The error description will be in the format: INVALID BLOCKSIZE/BLOCKING FACTOR CODE=nnn. The CODE=nnn translates directly into an IAM create abend code (see IAM user abend, error, and completion codes later in this section).

**Action:** If the message was issued for invalid blocking, ensure that the blocksize specified is both larger than the record length and compatible with the device. If the CODE= message is issued, refer to the actions for the associated IAM create abend code.

### IAM340 'recovery operation' - DSN= 'dsname'

**Reason:** Describes the user specified IAM file 'recovery operation' and the name of the IAM file being processed.

**Action:** None, information message only.

### IAM341 CONTROL BREAKS IN OVERFLOW BLOCKS -- = nnnn nnnn

**Reason:** Issued by the DIAGNOSE and RECOVER operations of the IAM file recovery program IAMRECVR, indicating that records within the overflow blocks of the file being processed are out of sequence.

**Action:** When issued during RECOVER, it indicates that the RECOVER output is out of sequence and must be sorted, either by IAMRECVR or externally prior to an IAM create. When issued during DIAGNOSE, no additional processing is necessary.

### IAM342 IAM FILE CONTAINS NO DETECTABLE ERRORS

**Reason:** Issued by the DIAGNOSE and RECOVER operations of the IAM file recovery program IAMRECVR, indicating that the IAM file is not damaged and is acceptable for IAM processing.

**Action:** None, information message only.
IAM343  **SEQUENCE ERRORS IN DATA BLOCKS -- = nnnnnnnn**

*Reason*: Issued by the DIAGNOSE and RECOVER operations of the IAM file recovery program IAMRECVR, indicating physical damage to the IAM file.

*Action*: The file must be created again from a suitable backup or recovered and reconstructed. Prior to create, sort the dataset output from the RECOVER operation.

IAM344  **NUMBER OF DROPPED BLOCKS -- = nnnnnnnn**

*Reason*: Issued by the DIAGNOSE and RECOVER operations of the IAM file recovery program IAMRECVR, indicating physical damage to the IAM file. IAMRECVR encountered I/O errors during the processing of the IAM file and nnnnnnnn blocks were dropped from the file.

*Action*: The file must be created again from a suitable backup or recovered and reconstructed. Prior to create, sort the dataset output from the RECOVER operation.

IAM345  **NUMBER OF DUPLICATE RECORDS -- = nnnnnnnn**

*Reason*: Issued by the DIAGNOSE and RECOVER operations of the IAM file recovery program IAMRECVR, indicating that duplicate keys were found in the IAM file. During a RECOVER operation, duplicate records can be ignored, printed, logged for subsequent application to the file, or applied directly if an IAM file is being created as the output of the recovery program.

*Action*: See the documentation for the RECOVER program, DUPLICATES operand. The file must be created again from a suitable backup or recovered and reconstructed. Prior to create, sort the dataset output from the RECOVER operation.

IAM346  **SPANNED RECORDS EXTRACTED --- = nnnnnnnn**

*Reason*: Issued by the IAMRECVR RECOVER command when spanned records have been recovered and placed in the dataset identified by the SPANOUT DD statement.

*Action*: To recover the spanned records, run the IAMRECVR APPLY SPANNED command, which will read the SPANOUT dataset, and insert the records back into the IAM dataset. See Section 45.03 for additional information.

IAM360  **STEP - sssssss DDNAME - 'ddname' DATASET MONITORED - 'dsname'**

*Reason*: Identifies the IAM job step name, the DDNAME and, optionally, the dataset name of the IAM file being processed by the information service routines of the IAM monitor facility.

*Action*: None, information message only.

IAM361  **INFO REQUESTED BY PROGRAM 'program' PERFORMING 'description' PROCESSING**

*Reason*: Identifies the IAM processing program that requested the printing of the information block, the type of processing being performed, and the time the information block was printed by the information service routines of the IAM INFO report.
Action: None, information message only.

IAM362 DATA CHARACTERISTICS

Reason: Heading line which always precedes the IAM data characteristics when listed by the IAMINFO report.

Action: None, information message only, for further detail on these fields see Section 10.72: IAM362.

IAM363 IAM FILE CHARACTERISTICS

Reason: Heading line which always precedes the IAM file characteristics when listed by the IAMINFO report.

Action: None, information message only, for further detail on these fields see Section 10.72: IAM363.

IAM364 IAM OVERFLOW CHARACTERISTICS

Reason: Heading line which always precedes the IAM overflow characteristics when listed by the IAMINFO report.

Action: None, information message only, for IAM Compatible format files only.

IAM365 IAM EXECUTION STATISTICS

Reason: Heading line which always precedes the IAM execution statistics when listed by the IAMINFO report.

Action: None, information message only, for further detail on these fields see Section 10.72: 365.

IAM366 IAM COMMAND EXECUTED SUMMARY

Reason: Heading line which always precedes the IAM command summary when listed by the IAMINFO report.

Action: None, information message only, for further detail on these fields see Section 10.72: IAM366.

IAM367 THERE WAS INSUFFICIENT STORAGE AVAILABLE TO ACQUIRE ADDITIONAL BUFFERS

Reason: IAM's Real Time Tuning tried to acquire additional data buffers but there was insufficient virtual storage available.

Action: Increase the REGION value on the JOB statement or the EXEC statement so the next execution of the job will have storage available to acquire additional buffers.
IAM368 SPECIFYING A BUFNO VALUE GREATER THAN \( nn \) MAY IMPROVE PERFORMANCE

**Reason:** IAM's Real Time Tuning buffer management found that, for this mix of data and file processing commands, if additional buffers had been available they would have been acquired. Additional buffers were not acquired because it would have exceeded the maximum buffers allowed for this job.

**Action:** If you wish to increase the maximum buffers for this job, specify the \( \text{MAXBUFNO}=nn \) parameter on the IAM Override Control statement (Section 11) for this IAM file.

*Note:* If the number of I/O's (EXCPs) for this file is relatively small, there is no real need to increase the maximum number of buffers.

IAM371 INCREASING THE NUMBER OF PRIME EXTENSION BLOCKS MAY IMPROVE PERFORMANCE

**Reason:** An attempt to insert a record into prime extension failed because the prime extension was full. The record was inserted into independent overflow instead.

**Action:** Define the IAM file with a larger prime extension.

IAM372 IAM EXTENDED AREA CHARACTERISTICS

**Reason:** This heading line always precedes the description of the IAM Extended Area (Enhanced format files only). See Section 10.72: IAM372.

IAM373 REORGANIZATION OF THIS FILE IS RECOMMENDED

**Reason:** During Open and/or Close processing, IAM noticed that the size of the extended overflow area was such that it could cause a performance degradation. As a preventative measure, it is recommended, but not required, that the file be reorganized to move records out of the overflow area. This IAMINFO message should be accompanied with an IAMW22 message, which will indicate the reason that a reorganization is being recommended.

**Action:** No action is required, but a file reorganization is recommended to be scheduled in the near future.

IAM400 'processing function' - 'progname' - VER v.r. - INNOVATION DATA PROCESSING - DATE: 'yyddd' PAGE: 'nn'

**Reason:** Generalized page header for the named 'processing functions' which are provided by the 'program' named in the printed title.

**Action:** None, information message only.

IAM401 PARM DATA - * parm-field-data *

**Reason:** Displays the program control information specified in the "PARM=" field of the EXEC statement. This data will not be displayed if program is invoked under TSO.

**Action:** None, information only.
IAM402 INVALID CONTINUATION

Reason: User coded a delimiting comma following the last keyword on a control statement and did not provide the next logical record.

Action: Correct and resubmit job.

IAM403 REQUIRED OPERAND(S) NOT SPECIFIED - cc...cc

Reason: The operand cc...cc is required for the execution of the command. It must be specified; no defaults are available.

Action: Correct and resubmit job.

IAM404 WORKING STORAGE AREA SIZE OF nn BYTES EXCEEDED - SUBCOMMAND IGNORED

Reason: The maximum number of working storage bytes available to the subcommands has been exceeded.

Action: The number of bytes available for subcommand working storage is derived from the 'MAXCORE' operand, which has a lower limit of 1000 bytes and an upper limit of 120000 bytes. If the upper limit has been reached, call INNOVATION for technical support. If the upper limit has not been reached, resubmit the job specifying a value for 'MAXCORE' greater than the number displayed.

IAM405 MAXIMUM CONTINUATION COUNT OF nnnn EXCEEDED - COMMAND FLUSHED

Reason: The user control statement used too many continuations.

Action: Reduce the number of continuations to the value nnnn. Resubmit the job.

IAM407 CONTROL STATEMENT ERROR - 'action taken'

Reason: An error was encountered during the processing of user supplied control statements. Always preceded by one or more messages which define and delimit the error(s). The 'action taken' is one of the following:

1. JOB TERMINATED - Processing will stop after the first error has been encountered.
2. SKIPPING FOR COMMAND - Processing will continue for all command statements within the SYSIN dataset.
3. RE-ENTER COMMAND OR END - Message when the SYSIN dataset is assigned to a TSO terminal. Re-enter command in error or 'END' to complete the processing.

Action: Correct and resubmit job.

IAM408 NO CONTROL STATEMENTS WERE FOUND JOB TERMINATED

Reason: SYSIN dataset is empty or contained only comment statements ("*" in column 1).
IAM410  DSNAME='dsname' -- 'error description' -- PROCESSING BYPASSED

**Reason:** The 'dataset name' printed encountered the 'error' described and subsequent processing was bypassed. The error description can be one of the following:

- NOT RECOGNIZABLE IAM FILE
- TRKCALC FUNCTION FAILED
- NOT KSDS VSAM
- FILE HAS ALTERNATE INDEX
- NAMEDS DATA NOT RETURNED
- UNABLE TO LOCATE DATA/INDEX
- KEYLEN GREATER THAN 250 BYTES
- RKP GREATER THAN 4096
- FILE WAS NEVER LOADED
- IS NOT VSAM OR IAM

**Action:** Check the dataset name for correctness. If the wrong dataset name was specified, specify the correct dataset name and resubmit the job. If the message indicates the dataset is not an IAM file it is because the IAM control record was not present. Use program IAMRECVR specifying the IAM file characteristics on the 'RECOVER' subcommand to facilitate data recovery.

IAM413  DSN/DSG INDEX ERROR - 'error description'

**Reason:** When using the index level option to select datasets or dataset groups the user either:

1. Specified too many index levels. (The maximum is 22).
2. Failed to provide significant data. (The leading periods were followed by a comma or blank).

**Action:** Correct and resubmit.

IAM418  COMMAND PROCESSING DETECTED ERROR -- 'action taken'

**Reason:** An error was encountered during the processing of the subcommand specified on a user supplied control statement. Always preceded by one or more messages that define and delimit the error(s). The action taken is one of the following:

- RE-ENTER COMMAND OR END - message for user when the SYSIN dataset is assigned to a TSO terminal. Re-enter command in error or 'END' to complete the processing.
- SKIPPING FOR COMMAND - processing will continue for all command statements found within the SYSIN dataset.
- REVERTING TO SYSIN - the error occurred when reading from an alternate command input source. Processing will continue for commands in the SYSIN dataset.

**Action:** Correct and resubmit job.
IAM421  LOCATE ERROR -- 'error description' - DSN='dsname'

**Reason:** A LOCATE SVC was issued requesting identification of the component named by DSN='dsname'. The LOCATE either failed or returned a component:cluster name type code that is not currently supported.

If the 'error description' is in the form: RETURN CODE rc - REASON IGGOCLAA - 'reason number' The LOCATE failed. Error codes are documented in the IBM System Messages Manual; under message number IDC3009I.

If the 'error description' is in the form: ENTYPE -- C(X'xx') The program encountered a type of component entry that it does not presently support.

If the 'error description' is in the form: NAMEDS -- C(X'xx') The program encountered a type of cluster entry that it does not presently support.

If the 'error description' is in any other form, an error exists in the catalog.

Note: This message only appears if DFEFERRPRT=YES had been setup earlier with IAMZAPOP or specified on the control statement. This is only a warning message. The program does not associate the component shown by DSN='dsname' with a cluster name, processing continues.

**Action:** Do one of the following:

For the RETURN CODE type of error, look up the codes under message IDC3009I. The most likely cause is a STEPCAT or JOBCAT DD statement was not supplied for a user catalog that has entries in it for components on the volume(s) being processed, and those entries are not ALIAS'ed.

For any other type of errors, contact INNOVATION technical support for assistance.

IAM465  UNABLE TO OPEN (DDNAME=DSN=)'name' - 'reason'

**Reason:** The 'ddname' named in the message failed to OPEN for the 'reason' shown in the text. The recognized reasons are:

- ABEND exit taken.
- VOL='volser'
- denied by OPEN exit.
- DD='ddname' statement missing/ misspelled or incorrectly specified.
- RC=nn ERROR CODE=nnn

**Action:** If the named 'ddname' is required for the requested operation correct the error and resubmit the job, otherwise the message may be ignored.
IAM471  DDNAME='ddname' I/O ERROR -- SYNAD='message'

Reason: A permanent I/O error was detected on the dataset referenced by the DDNAME 'ddname'.

Action: Examine the SYNADAF message to determine the cause of the error. The format of this message is documented in IBM SRL publications. Call INNOVATION for additional assistance.

IAM476  DDNAME='ddname' REFERENCES A DEVICE TYPE OTHER THAN DISK

Reason: The DDNAME 'ddname' listed can only be allocated to a disk device.

Action: Check the unit specification in the JCL for errors, correct and resubmit.

IAM484  INTERNAL LOGIC ERROR -- JOB TERMINATED

Reason: The program has encountered an illogical condition.

Action: Call INNOVATION for additional assistance AFTER obtaining a storage dump.

IAM485  SORT FAILURE HAS OCCURRED RC='nn' - 'action description'

Reason: Your installation's sort product has set a return code other than zero. The return code problem description can be found in the sort manual supplied with the sort program product. In many cases the sort program will provide an error message on DDNAME SYSOUT and or on the system console.

Action: If you cannot correct the problem from this information, call INNOVATION for additional assistance.

IAM491  'function' FUNCTION STARTED TIME=hh.mm.ss

Reason: Identifies type of function and time the function started.

Action: None, information message only.

IAM492  'function' FUNCTION ENDED -- hh.mm.ss CONDITION CODE - nnnn

Reason: Identifies type of function and time the function ended. The return code is printed if it is non-zero.

Action: Review the completion code. If the completion code is greater than zero, check the output for the preceding messages that describe the reason for a non-zero completion.

IAM493  'function' - BYPASSED -- PRIOR FUNCTION TERMINATED WITH KEYWORD OR COMMAND DETECTED ERROR

Reason: A previous command upon which this function may be dependent encountered serious errors.
IAM495  NO RECORDS MATCHED SELECTION CRITERIA

**Reason:** The selection criteria specified did not cause any records to be selected for processing. If program IAMSMFVS, this message will also appear if the required SMF record types are not being collected.

**Action:** Make sure the selection criteria is correct.

If IAMSMFVS: Make sure you are collecting SMF records types 4 or 30 subtype 4 and type 64. If you are running against an SMF history tape, make sure that the required record types are being copied to the history tapes.

If IAMSMF: Make sure you are collecting SMF records types 4 or 30 subtype 4 and type 14, 15, 64. If you are running against an SMF history tape, make sure that the required record types are being copied to the history tapes.

IAM496  MODULE 'modname' -- NOT USABLE WITH RELEASE v.r. PROGRAMS -- EXECUTION TERMINATED

**Reason:** Module 'modname' is a release/version that is incompatible with the load module being executed.

**Action:** Check for a STEPLIB/JOBLIB DD statement pointing to a library other than the correct library for the product level you expect to use. If present, correct the library name and resubmit the job. If there is no STEPLIB/JOBLIB present, the module 'modname' is being obtained from a LINKLIST library. Add a STEPLIB or JOBLIB DD statement specifying the correct load module library and resubmit the job.

IAM497  cc..cc ABNORMALLY TERMINATED DUE TO KEYWORD/COMMAND DETECTED ERRORS

**Reason:** The common parsing routine encountered errors in parsing the user specified control statements. In addition, the user has set the option KWDCC=ABEND via program IAMZAPO. Always preceded by one or more error description messages.

**Action:** Previous message(s) describe the error(s); see those messages for further details.

IAM498  cc...cc PROCESSING COMPLETED WITH ERRORS

**Reason:** The named program completed the requested processing but encountered abnormalities in the process.

**Action:** Check the output for preceding messages that may describe the errors in detail.

IAM499  cc...cc PROCESSING COMPLETED

**Reason:** The named program has completed processing as requested.

**Action:** Check the output for messages that describe the results.
IAM516  CAMLST REGISTERS R0=nnnnnnnn R1=nnnnnnnn R15=nnnnnnnn

**Reason:** A CAMLST request failed. The type of CAMLST function (shown in the immediately preceding message) and the registers make diagnosing the problem relatively simple. The return code(s) from catalog management is(are) documented in the IBM SRL SYSTEM PROGRAMMING LIBRARY: DATA MANAGEMENT (for MVS) or CATALOG ADMINISTRATION GUIDE (for MVS/XA) or DATA MANAGEMENT FOR SYSTEM PROGRAMMERS (for non-MVS).

**Action:** Check the return code(s) and take corrective action if the error is apparent or call INNOVATION for additional assistance.

IAM530  MODULE 'modname' NOT FOUND - 'ddname' - 'dsname'

**Reason:** A BLDL was issued for the module 'modname' in the dataset 'dsname' referenced by 'ddname' and the module was not found. The module is required to support a requested processing function.

**Action:** Verify that the library dataset name 'dsname' specified on the 'ddname' statement was correct. If incorrect, correct it and resubmit the job. If the dataset name is correct, check the IAM product install listing to see that all steps executed successfully. If required, rerun the installation steps missed. If necessary contact INNOVATION for additional assistance.

IAM531  MODULE 'modname' CONTAINS NO TEXT RECORDS - 'ddname' - 'dsname'

**Reason:** When an attempt was made to read the module 'modname' in the dataset 'dsname' referenced by 'ddname' and no text records were found. The module is required to support a requested processing function.

**Action:** Verify that the library dataset name 'dsname' specified on the 'ddname' statement was the correct one. If incorrect, change and resubmit the job. If the dataset name is correct, check the IAM product install listing to see that all steps executed successfully. If required, rerun the installation steps missed. If necessary contact INNOVATION for additional assistance.

IAM532  MODULE 'modname' I/O ERROR READING - 'ddname' - 'dsname'

**Reason:** When an attempt was made to read the module 'modname' in the dataset 'dsname' referenced by 'ddname' an I/O error was encountered. The module is required to support a requested processing function.

**Action:** Check the SYNAD error message(s) and MINI DUMP produced to see if the cause of the error is an obvious one. If so, correct and resubmit the job. If necessary contact INNOVATION for additional assistance.

IAM533  MODULE 'modname' I/O ERROR WRITING - 'ddname' - 'dsname'

**Reason:** When an attempt was made to write the module 'modname' in the dataset 'dsname' referenced by 'ddname' an I/O error was encountered.

**Action:** This is a serious error which may result in the load module library referenced by 'dsname' being unusable. If necessary, restore or recreate the library. Check the SYNAD error message(s) and MINI DUMP produced to see if the cause of the error is
an obvious one. If so, correct and resubmit the job. If necessary contact INNOVATION for additional assistance.

IAM534  'ddname' OPEN FAILURE -- AVAILABLE COMMANDS: HELP, END

Reason: The required 'ddname' statement could not be OPENed. Commands which access modules in the 'ddname' dataset cannot be executed.

Action: If the 'ddname' dataset is required for the operation being attempted, correct and resubmit the job. Otherwise, none, information only.

IAM535  MODULE 'modname' READ UNSUCCESSFUL -- 'command' BYPASSED

Reason: The user requested function 'command' be performed against module 'modname' but the module was not successfully read. This message will be preceded by message IAM530, IAM531, or IAM532 detailing the reason the read failed.

Action: See message IAM530, IAM531, or IAM532, as required.

IAM538  MODULE 'modname' -- WRONG VER/LEVEL 'ddname' - 'dsname'

Reason: Module 'modname' is a release/version that is incompatible with the load module being executed.

Action: Verify that the 'ddname' DD statement is pointing to the correct library for the product level you expect to use. If incorrect, change and resubmit the job. If the dataset name is correct, check the IAM product install listing to see that all steps were executed successfully. If required, rerun the installation steps missed. If necessary, contact INNOVATION for additional assistance.

IAM539  MODULE 'modname' 'function' - 'ddname'- 'dsname'

Reason: The module 'modname' from the dataset dsname' referenced by 'ddname' has had function 'function' performed as requested by the user.

Action: None, information only.

IAM540  cccccccc CONTAINS INVALID CHARACTERS -- ZAP REJECTED

Reason: The value specified for the operand 'ccccccc' contained one or more characters that were not A-Z, 0-9, $#@.

Action: Remove the invalid character(s) and resubmit the job.

IAM541  THE FOLLOWING OPTIONS HAVE BEEN CHANGED IN -- 'dsname'

Reason: The AUDIT function has found changed options in the options table. The changed options will be listed.

Action: N/A
IAM542  cccccc CONTAINS INVALID INDEX STRUCTURE -- ZAP REJECTED

**Reason:** The value specified for the operand cccccc contains two (2) or more consecutive periods (..) in violation of operating system dataset naming conventions.

**Action:** Correct the error(s) and resubmit the job.

IAM543  cccccc DOES NOT CONTAIN CHARACTER STRING cccccc -- ZAP REJECTED

**Reason:** The value specified for the operand cccccc does not contain the character string cccccc as the required start of an index level.

**Action:** Correct the error(s) and resubmit the job.

IAM544  cccccc REJECTED -- EXCEEDS MODIFIABLE PORTION OF IAMOPT

**Reason:** The length of the value specified for the operand cccccc taken in conjunction with the offset operand exceeds the modifiable portion of the IAMOPT table.

**Action:** Correct the error(s) and resubmit the job.

IAM545  VERIFY FAILED -- CHAR/HEX PRINT FORCED

**Reason:** The VERIFY of existing contents failed. A character/hexadecimal print of the module IAMOPT is produced. Always preceded by message IAM544.

**Action:** See message IAM544.

IAM546  AUDIT REQUEST COMPLETE ALL DEFAULTS SET IN -- 'dsname'

**Reason:** The AUDIT function has found no changed options in the options table.

**Action:** N/A

IAM547  AUDIT REQUEST COMPLETE FOR -- 'dsname'

**Reason:** The AUDIT function has completed. Changes were found and listed.

**Action:** N/A

IAM550  ERRORS ENCOUNTERED DURING EXECUTION -- REWRITE CANCELED

**Reason:** One or more modules from the SYSLIB dataset were scheduled for rewrite at termination or by the execution of a REWRITE command. However, previous commands failed to complete successfully. The errors encountered have been documented by preceding error messages.

**Action:** Correct the error conditions documented by the preceding error messages and resubmit the job.
IAM561 NON-NUMERIC DATA VALUE SPECIFIED FOR THE [FROMDATE | TODATE] KEYWORD

Reason: An invalid character string was specified for the indicated keyword.

Action: Correct the error condition. For information on valid values, refer to the IAMSMFVS Section 40.03 of the manual.

IAM562 INVALID VALUE SPECIFIED FOR [FROMDATE | TODATE]

Reason: An invalid numeric string was specified for the indicated keyword.

Action: Correct the error condition. For information on valid values, refer to the IAMSMFVS Section 40.03 of the manual.

IAM569 WARNING - cccccccc NOT CATALOGED - DSN='dsname'

Reason: The dataset 'dsname' indicated is the new default for the file type indicated by cccccccc. A CAMLST LOCATE was issued to verify that the dataset is cataloged as required for IAM to dynamically allocate. The dataset name was not found by LOCATE. Message IAM516 detailing the CAMLST return code(s) is always printed following this message.

Action: See message IAM516. This is only a warning message. The dataset name listed has become the new name of the indicated default.

IAM574 'request' - DDNAME='ddname' - 'action'

Reason: Failed I/O 'request' resulting in the named 'action':

- ** DD NOT OPEN - EXTRACT REQUEST DENIED
- GET FAILED - EOF FORCED
- CLOSE FAILED - JOB TERMINATED
- OPEN FAILED - COPY REQUEST DENIED
- OPEN FAILED - EXTRACT REQUEST DENIED
- OPEN FAILED - REPORT REQUEST DENIED
- ** OPEN FAILED - REQUEST DENIED
- OPEN FAILED - TO DDNAME COPY DENIED
- RJFCB FAILED - REQUEST DENIED

The 'ddname' shown in the message is required as either INPUT TO or OUTPUT FROM a processing program. An unexpected failing request for an I/O related service against that DDNAME has forced the program to take the remedial action shown in the message text.

Action: Correct the reason for the error and resubmit the job.
IAM600  MAXSTACK VALUE OF nnnnnnn EXCEEDED - SMF RCD PROCESSING TERMINATED

Reason: The SMF record processing subcommand being executed utilizes an in storage stack specified by the operand MAXSTACK. This storage stack is full. The specified size is insufficient to allow all of the selected SMF records to be processed. IAMSMF terminates without producing a report.

Action: Re-execute the program and specify the operand MAXSTACK= with a value greater than the value nnnnnn printed in this message.

IAM601  SMF RECORDS -- READ.nnnnnnnn USED.nnnnnnnn DROPPED.nnnnnnnn

Reason: Documents the SMF records processed as follows:

READ  - number of SMF records read from the input dataset. May not reflect the total number of records in the dataset if an IAM600 message was issued.

USED  - number of SMF records selected from the input dataset by the criteria specified by the user.

DROPPED- number of SMF records dropped from the input dataset because of length checking, or other user specified criteria.

Action: None, information only.

IAM602  FOLLOWING SMF RECORD DROPPED - LENGTH CHECK

Reason: User specified the CHECKLENGTH operand, causing all SMF records selected to be checked against a table of minimum lengths. This record failed the length check. The first 32 bytes of the record are printed in hexadecimal.

Action: Use the keyword RECSIZE=nn on the report control statement to specify an appropriate size.

IAM603  VSAM ERROR -- (DDNAME= DSN=)'name' 'error description' -- R15=xxxxxxxx - CODE=xxxxxxxx

Reason: An error occurred during the processing of a VSAM dataset. The error description will be one of the following:

- CLOSE failed -- close of VSAM dataset.
- GENCB failed -- generation of a control block.
- GET failed -- get next record.
- OPEN failed -- OPEN of VSAM dataset.

Action: Check the values of R15 and CODE against the return codes listed in the 'VSAM REFERENCE FOR MVS/370 DFP', 'VSAM ADMINISTRATION: MACRO INSTRUCTION REFERENCE FOR MVS/XA', or 'OS/VS VIRTUAL STORAGE ACCESS METHOD PROGRAMMER'S GUIDE', to determine the cause of the error. If possible, correct and re-execute the job. Call INNOVATION for additional assistance.
IAM619  MODULE 'modname' TOO LARGE TO PROCESS -- 'ddname' - 'dsname'

 Reason: IAMZAPOP attempted to read the module 'modname' in the dataset 'dsname' referenced by the statement 'ddname'. The module was required to support a processing functions but, due to the size of the module (or previously read modules), not enough buffer storage was available to complete the read.

 Action: Run IAMZAPOP specifying 'ZAP BUFSIZE=nnnn', where nnnn is the size of the buffer in bytes. Specify a value between 122,880 and 3,145,728. After the buffer size has been changed, rerun the job that failed.

IAM620  MAXJOB VALUE OF nnnnn EXCEEDED - ADDITIONAL JOB NAMES BYPASSED

 Reason: The maximum number of unique job names tabled by IAMSMFVS has been exceeded. SMF records that match the specified selection criteria but with job names other than those already tabled will be bypassed and not reported.

 Action: Specify a MAXJOB value greater than nnnnn but less than or equal to32,000 and, if required, rerun the job.

IAM621  MAXDSN VALUE OF nnnnn EXCEEDED - ADDITIONAL DATASET NAMES BYPASSED

 Reason: The maximum number of unique dataset names tabled by has been exceeded. Records that match the specified selection criteria but with dataset names other than those already tabled will be bypassed and not reported or will be shown in a second report if present.

 Action: Specify a MAXDSN value greater than nnnnn but less than or equal to 32,000 and, if required, rerun the job.

IAM622  MAXDSN VALUE OF nnnnn EXCEEDED - ADDITIONAL VSAM CLUSTER RELATIONSHIPS BYPASSED

 Reason: IAMSMFVS issues a LOCATE for each ICF/VSAM DATA and/or INDEX component and adds the resulting cluster name to the tabled dataset names. Those DATA and/or INDEX components that remain unrelated (I.E.: have not had the cluster name appended) when the value of MAXDSN is exceeded will be printed as unique datasets.

 Action: ICF/VSAM clusters can take up to three (3) entries in the dataset name table that is generated from the MAXDSN operand. If you only plan to extract information on ICF/VSAM clusters and you expect to have approximately 1000 clusters listed, specify a MAXDSN no less than 3000. If the error reoccurs, specify a MAXDSN value greater than nnnnn but less than or equal to 32,000 and, if required, rerun the job.

IAM634  SEQUENCE CHECKS FORCE SORTING OF DATA BEFORE IAM CREATION

 Reason: The file recovery program found the records it was processing were not in ascending sequential order. An IAM file creation expects the IAM file to be loaded in order. The set of keys the recovery program is processing cannot be used as is to load an IAM file. The Independent Overflow records in an IAM file may not be in collating sequence. Recovery of a file with Independent Overflow could result in this message.
**Action:** The recovered records must be sorted before they can be used to load an IAM file.

**IAM635**  
**LOG AND IAM FILE ARE INCOMPATIBLE**

**Reason:** The file recovery program found the log file records it was applying were not compatible with the IAM file it was rebuilding.

**Action:** The log file being applied and the IAM file in question should be checked to see if they actually represent the same data before attempting to continue with a file recovery.

**IAM636**  
**APPLY FAILED -- RECORD KEY FOLLOWS --**

**Reason:** The file recovery program found the record it was applying was not compatible with the IAM file it was rebuilding. **Action:** The log file being applied and the IAM file in question should be checked to see if they actually represent the same data before attempting to continue with a file recovery.

**IAM637**  
**IAM INTERNAL BLOCK --**

**Reason:** The block shown contains file control information.

**Action:** None, information message only.

**IAM638**  
**IAM BLOCK NUMBER nnnnnn --**

**Reason:** The block shown is relative block number 'nnnnnn' in the file.

**Action:** None, information message only.

**IAM700**  
**Innovation Access Method Trace VER nnnnnnnn Date:yyyy.ddd Page:nnn**

**IAM701**  
**DDNAME:ddname  DSN:dataset name**

**IAM702**  
**Type  Time  Request  RPL/RBN  PLH  OPTCD RC/RINFO RECLEN  Key**

**IAM703**  
****  ****  --------  --------  ---  ----- -------- ------  ---

**Reason:** IAM700 - IAM703 are headings for IAM trace output.

**IAM704**  
**Trace Ended**

**Reason:** indicates the end of IAM tracing.

**IAM705**  
**There were trace lock failures**

**Reason:** If any trace lock failures occurred during processing, this message will be issued.

**IAM710**  
**IOS 08.33.38.1900 PUT  00023A10 00040298 2040 01FE 00002D00**

**Reason:** This is the I/O Start trace detail line. The contents of each field are as follows:
IAM SYSPRINT Messages

- Type     - IOS Request - One of the following, depending on the type of I/O requested: GET, PUT, POINT, ERASE, CHECK or ENDREQ.
- Time     - Time the trace record was generated.
- RPL/RBN  - The address of the RPL used to request the I/O.
- PLH      - The address of the active PLH for this I/O.
- OPTCD    - The values of RPLOPTCD1 and RPLOPTCD2 at the time of the request.
- RC/RINFO - blank
- RECLEN   - The requested record length (if applicable).
- Key      - The records RBA or key (if applicable).

Notes: When trace is activated on a PATH, the trace output is modified to include a suffix to the message number. When the trace entry is for a path, the suffix is “P”, for the alternate index the suffix is “X”, and for the base cluster, the suffix is “B”

IAM711  IOE 08.33.39.2200 PUT 00023A10 00040298 0000 01FE 0005A000

Reason: This is the I/O End trace detail line. The contents of each field are as follows:

- Type     - IOE Request - One of the following, depending on the type of I/O requested: GET, PUT, POINT, ERASE, CHECK or ENDREQ.
- Time     - Time the trace record was generated.
- RPL/RBN  - The address of the RPL used for this request.
- PLH      - The address of the active PLH for this I/O.
- OPTCD    - N/A
- RC/RINFO - The return code from the I/O.
- RECLEN   - The record length (if applicable).
- Key      - The records RBA or key (if applicable)

Notes: When trace is activated on a PATH, the trace output is modified to include a suffix to the message number. When the trace entry is for a path, the suffix is “P”, for the alternate index the suffix is “X”, and for the base cluster, the suffix is “B”

IAM712  BFR 08.33.39.2300 WRITE 00000026 00040298 1008 0005A1FE

Reason: This is the Buffer Manager trace detail line. The contents of each field are as follows:

- Type     - BFR Request - One of the following, depending on the type of processing required: READ, WRITE, RELEASE, FLUSH.
- Time     - Time the trace record was generated.
- RPL/RBN  - The Relative Block Number (RBN) of the requested data block.
- PLH      - The address of the active PLH for this I/O.
- OPTCD    - PLH option bytes 1/2.
IAM713 EXCP 08.33.39.2400 READ 00000024 00040298 0001

Reason: This is the EXCP trace detail line. The contents of each field are as follows:

- TYPE       - EXCP Request    - One of the following, depending on the type of processing required:
  - PRFMT - Preformat additional blocks
  - WREOF - Write EOF
  - WRADD - Write new block
  - READ - Read single or multiple blocks
  - WRITE - Write single or multiple blocks
- Time       - Time the trace record was generated.
- RPL/RBN    - Relative Block Number (RBN) of first block processed.
- PLH        - The address of the active PLH for this I/O.
- OPTCD      - Number of blocks to process.
- RC/RINFO   - N/A.
- RECLNE     - N/A.
- Key        - N/A.

IAM714 XTND 08.34.02.0100 EXTEND 00000028 00000000 0102 0000 20980001

Reason: This is the XTND trace detail line. The contents of each field are as follows:

- TYPE       - XTND Request    - One of the following, depending on the type of EXTEND processing required:
  - EXPAND - Expand into allocated but unused area.
  - EXTEND - Obtain new extent.
  - EXP/EXT - Expand and obtain new extent.
- Time       - Time the trace record was generated.
- RPL/RBN    - RBN of first new block after EXTEND.
- PLH        - The address of the active PLH for this I/O.
- OPTCD      - Volume number and extent number of new extent.
- RC/RINFO   - Return code from EXTEND processing.
- RECLNE     - N/A.
- Key        - EXTEND processing internal flags.

IAM715 hh.mm.ss.nnnn PUT xxxxxxxx oooooooo rrrrrrrr llllll kkkkkkkkkkkkkkkk

Reason: This is the file load detail trace line. The contents of each field are as follows:
IAM SYSPRINT Messages 80.03

- 80-53 -

• **hh.mm.ss.nnnn** – Time the request was issued.

• **PUT** – VSAM I/O request type. Normally for file load, this will only be a PUT. Other types of I/O requests (i.e., GET) are invalid during a file load, although if such a request is attempted it will be included on the trace.

• **xxxxxxxx** – The address of the Request Parameter List (RPL), in hexadecimal.

• **oooooo0o** – The 4 bytes, in hexadecimal, of the RPL OPTCD field.

• **rrrrrrrr** – The return code and logical error code, in hexadecimal, from the RPLFDBWD field.

• **llllllll** – The length of the record being written to the file, from the RPLRLEN field.

• **kkkkkk...** - The key of the record being written. For ESDS files, this will be the RBA of the record generated by IAM. The key will be displayed in either hexadecimal, or in character, depending on the name of the TRACE DD statement. The key will be continued on subsequent lines, if necessary.

IAM716  +kkkkkkkk

**Reason:** This is the continuation of the key of the record being processed, from the preceding IAM715 message.
80.04 IAM Journal Exit WTO Messages

IAMJ01 IAMDD ddname: INITQ BROKEN, JOURNAL PROCESSING TERMINATED.

**Reason:** An error occurred during the IAM Journal Exit initialization processing. This error can occur when multiple concurrent I/O requests are active and are requiring service from the IAM Journal Exit while it is attempting to allocate and open the log dataset.

**Action:** I/O requests will continue to be serviced to the IAM dataset, however journaling is no longer being done by IAM for the indicated IAM dataset. Contact Innovation for assistance.

IAMJ02 IAMDD ddname: ALLOCATION OF LOG FILE FAILED, JOURNAL PROCESSING TERMINATED.

**Reason:** The dynamic allocation of the log file for the specified IAM dataset has failed.

**Action:** I/O requests will continue to be serviced for the indicated IAM dataset, however journaling is not being done. If you are having difficulty determining why the LOG dataset is not able to be allocated, contact Innovation for assistance.

IAMJ03 IAMDD ddname: GETMAIN1 FAILED, JOURNAL PROCESSING TERMINATED.

**Reason:** There is insufficient below the 16 megabyte line storage available in the region for IAM to acquire the storage required to handle the requested journaling. The amount of storage being requested that resulted in this failure is generally less than 2K. The OPEN of the indicated IAM dataset will be failed.

**Return Codes:** A reason code of 188(x'BC') is stored in the ACB error flags field (ACBERFLG) and the open request is failed with a return code of 8.

**Action:** Generally increasing the REGION parameter for the job step should correct this problem. If it does not, then contact Innovation for assistance. This problem can be circumvented by turning journaling off for this file, through the use of the IAM overrides. (i.e., JRNAD=NONE).

IAMJ04 IAMDD ddname: OPEN OF LOG FILE FAILED, JOURNAL PROCESSING TERMINATED.

**Reason:** The OPEN of the log file has failed. There should be some IBM messages indicating the cause of the failure.

**Action:** I/O requests will continue to be serviced for the indicated IAM dataset, however journaling is not being done. Correct the error condition based on the information available.

IAMJ05 IAMDD ddname: BUFFER GETMAIN FAILED, JOURNAL PROCESSING TERMINATED.

**Reason:** There is insufficient storage available to obtain the necessary I/O buffers for the IAM journal processing. For systems that are at a high enough level of DFSMS, the
buffers are requested above the 16 megabyte line, otherwise the storage is requested from below the 16 megabyte line.

**Action:** I/O requests will continue to be serviced for the indicated IAM dataset, however journaling is not being done. The journal exit requests storage for five (5) buffers, so multiply the block size of the dataset by 5 to determine storage requirements, and adjust your region parameter as necessary.

**IAMJ06 IAMDD ddname: SYNAD MESSAGE GOES HERE..........................**

**Reason:** An I/O error has occurred on the IAM Log dataset. Review the I/O error cause from the message displayed.

**Action:** I/O requests will continue to be serviced for the indicated IAM dataset, however journaling is not being done. Correct the problem causing the I/O error.

**IAMJ08 IAMDD ddname: LOCATE FAILED FOR THE LOG FILE, JOURNAL PROCESSING TERMINATED.**

**Reason:** In an attempt to prevent allocation and open errors, the IAM Journal exit program will verify that the required log file is in the catalog. If it is not found in the catalog, then the OPEN request for the IAM dataset will also fail.

**Return Codes:** A reason code of 188(x'BC') is stored in the ACB error flags field (ACBERFLG) and the open request is failed with a return code of 8.

**Action:** Make sure that the required log dataset is properly allocated and cataloged. Specifying the IAM ACCESS override JRNAD=NONE can be used to turn off journaling until the cause of the error is found.

**IAMJ09 IAMDD ddname: OBTAIN FAILED FOR THE LOG FILE, JOURNAL PROCESSING TERMINATED.**

**Reason:** The IAM journal exit will attempt to make sure that the required log dataset is allocated on the volume indicated by the catalog prior to attempting an allocation and open of this critical file. If the required log dataset is not found, then the OPEN for the IAM dataset will be failed.

**Return Codes:** A reason code of 188(x'BC') is stored in the ACB error flags field (ACBERFLG) and the open request is failed with a return code of 8.

**Action:** Make sure that the required log dataset is properly allocated and cataloged. Specifying the IAM ACCESS override JRNAD=NONE can be used to turn off journaling until the cause of the error is found.

**IAMJ10 IAMDD ddname: DCB ATTRIBUTES OF LOG FILE ARE INCOMPATIBLE, JOURNAL PROCESSING TERMINATED.**

**Reason:** The IAM journal exit will make sure that the attributes (RECFM, LRECL, and BLKSIZE) are appropriate as needed by the file being logged. If they do not meet the required criteria, the OPEN of the IAM dataset will fail.

**Return Codes:** A reason code of 188(x'BC') is stored in the ACB error flags field (ACBERFLG) and the open request is failed with a return code of 8.
**Action:** Review the documentation on the IAM journaling exit to make sure that the DCB attributes selected for the LOG dataset are appropriate, and change them as necessary. It is best to let IAM determine the DCB attributes, by just allocating the DASD space required for the LOG dataset without specifying any DCB attributes.

**IAMJ1** IAMDD ddname: SAVEAREA GETMAIN FAILED, JOURNAL PROCESSING TERMINATED.

**Reason:** There is insufficient virtual storage available for the IAM Journal Exit. The OPEN of the IAM dataset will be failed. The amount of storage being requested is generally less than 1K, and it can reside in either above or below the 16 megabyte line.

**Return Codes:** A reason code of 188(x'BC') is stored in the ACB error flags field (ACBERFLG) and the open request is failed with a return code of 8.

**Action:** Increase the REGION parameter of the job step that is failing with this error.
80.05 IAM/RLS and IAM/PLEX Messages

The following informational and error messages may occur during IAM/RLS or IAM/PLEX processing. They will generally appear on the RLSLOGDD, and some critical error message will appear on the consoles and system logs. For most of the information provided on the messages, the use of IAMRLS or IAM/RLS generally can also occur for IAM/PLEX address spaces. Several of the messages are for diagnostic purposes that will contain information useful only to Innovation technical support personnel.

IAML0001 LOGGING OF MESSAGES DISABLED

Reason: IAMRLS was unable to obtain storage for the message table used by IAMLOGER and has disabled logging of messages to RLSLOGDD.

Action: Increase the REGION size of the IAMRLS address space.

IAML0002 GLOBAL NAME TOKEN AREA RETRIEVED

Reason: IAMRLS was able to retrieve and reuse the Global Token area left from a previous iteration of the IAMRLS address space.

Action: None

IAML0003 UNABLE TO START VIF

Reason: IAMRLS received a non-zero return code from IAMSTART while attempting to start VIF.

Action: Check for other messages issued by IAMSTART to the SYSLOG which will provide further information as to the reason VIF could not start.

IAML0004 GETMAIN OR LOCK OBTAIN FAILURE

Reason: IAMRLS could not obtain the local lock needed for a CSA getmain, or there was not enough CSA available to satisfy the CSA getmain request.

Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0005 GLOBAL NAME TOKEN AREA CREATE FAILED

Reason: IAMRLS was unable to create a Global Token area.

Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0006 DATASPACE BUILD FAILED

Reason: IAMRLS was unable to create a data space.

Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0007 SYSEVENT FAILED

Reason: The SYSEVENT macro issued to make the IAMRLS address space non-swappable failed.
Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0008 LOAD OF SERVICE MODULE FAILED

Reason: IAMRLS was unable to load one of its service modules.

Action: Make sure you have an IAM load library that has all of the modules distributed on the IAM distribution tape included in either the LINKLIST or a JOBLIB/STEPLIB.

IAML0009 LOCAL NAME TOKEN AREA CREATE FAILED

Reason: IAMRLS was unable to create the Local Name Token area.

Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0010 GETMAIN OF CELL POOL STORAGE FAILED

Reason: IAMRLS was unable to acquire storage for the cell pools set up during initialization.

Action: Raise the region size for the IAMRLS address space.

IAML0011 TASK WORK AREA GETMAIN FAILED

Reason: The getmain for the TASK work area storage failed.

Action: Raise the region size for the IAMRLS address space.

IAML0012 ATTACH PROCESSOR FAILED, CANNOT CONTINUE

Reason: IAMRLS was unable to attach one of its subtasks.

Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0013 GLOBAL NAME TOKEN AREA DELETED

Reason: IAMRLS deleted the Global Token Area during shutdown.

Action: None.

IAML0014 OPEN/CLOSE TASK ATTACHED

Reason: IAMRLS attached the OPEN/CLOSE subtask.

Action: None.

IAML0015 LOGGING TASK ATTACHED

Reason: IAMRLS attached the LOGGING subtask.

Action: None.
IAML0016  WORKLOAD MANAGER ATTACHED

Reason: IAMRLS attached the WORKLOAD MANAGER subtask.

Action: None.

IAML0017  JOURNAL TASK ATTACHED

Reason: IAMRLS attached the JOURNAL subtask.

Action: None.

IAML0018  TRACE TASK ATTACHED

Reason: IAMRLS attached the TRACE subtask.

Action: None.

IAML0019  IAMRLS ESTAE ENTERED FOR ABEND SxxX

Reason: The ESTAE for module IAMRLS was entered for an abend condition.

Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0020  PRELOAD OF PROCESSING MODULE FAILED

Reason: IAMRLS was unable to preload some of the processing modules.

Action: Check the IAM load library to make sure it contains all of the modules distributed with the IAM product.

IAML0021  ADDRESS SPACE ALREADY ACTIVE

Reason: IAMRLS detected an address space running under the same name.

Action: None.

IAML0022  IAMBREST SUBTASK ATTACHED

Reason: IAMRLS attached the DYNAMIC BACKOUT subtask.

Action: None.

IAML0023  IAMBWJRN SUBTASK ATTACHED

Reason: IAMRLS attached the Journal Buffer Flush subtask.

Action: None.

IAML0024  cccccccc SUBTASK ATTACHED

Reason: The subtask named cccccccc in the message was attached during startup.

Action: None.
IAM/RLS and IAM/PLEX Messages

IAML0025  BASE NAME TOKEN AREA CREATE FAILED

**Reason:** IAMRLS was unable to create a Global Token area to be used as an anchor for the Multiple IAMRLS address space support.

**Action:** Contact Innovation for help in diagnosing the cause of this problem.

IAML0026  IAMRLS ACTIVE WITH RLSID= nnnn

**Reason:** IAMRLS address space is active, with the indicated RLSID.

**Action:** None.

IAML0027  RLSID OF TEST IS INVALID, CANNOT CONTINUE

**Reason:** Specification of an RLSID value of TEST is not permitted on the start up of IAM/RLS or IAM/PLEX. An RLSID of TEST is automatically assigned when an IAM/RLS address space is started with a PARM=TEST on the start command or on the EXEC card in the proc.

**Action:** Change the value of the RLSID parameter in the startup parameters for the IAM/RLS or IAM/PLEX proc to something other than TEST. If you desire to start a TEST IAM/RLS address space with an RLISID of TEST, then start it with PARM=TEST.

IAML0028  IAMRLS ACTIVE WITH RLSID=rlsid IS A REMOTE RLS ONLY

**Reason:** Indicates that the IAM/PLEX address space was started with the REMOTERLS parameter. This IAM/PLEX address space can only handle file I/O operations, it cannot be directly connected to users of the IAM/PLEX facilities. This is used when multiple IAM/PLEX address spaces in the same RLSGROUP are active on the same LPAR.

**Action:** None, informational only message.

IAML0029  BRLOK INITIALIZATION FAILED, CANNOT CONTINUE

**Reason:** Indicates that the IAM Record Lock Manager initialization has failed. There should be additional messages indicating the exact cause of this failure.

**Action:** Review the other messages and correct the error condition, then restart the IAM/RLS or IAM/PLEX address space. For assistance, contact Innovation technical support.

IAML0030  RESOURCE MANAGER ADD FAILED, IAMRESMG NOT ACTIVE AS IAMRLS RESOURCE MANAGER

**Reason:** Unable to activate IAMRESMG as the IAMRLS resource manager

**Action:** IAMRLS will be unable to cleanup resources for failing jobs or address spaces. Contact Innovation for support.
<table>
<thead>
<tr>
<th>Message ID</th>
<th>Message Content</th>
</tr>
</thead>
</table>
| IAML0031   | IAMRESMG NOW ACTIVE AS IAMRLS RESOURCE MANAGER  
**Reason:** IAMRESMG has been activated as the IAMERLS resource manager.  
**Action:** None. |
| IAML0032   | DELETING IAMRESMG AS RESOURCE MANAGER  
**Reason:** IAMRESMG is being deleted as the IAMRLS resource manager.  
**Action:** None. |
| IAML0033   | IAMRESMG DELETING AS RESOURCE MANAGER.  
**Reason:** IAMRESMG has been deleted as the IAMRLS resource manager.  
**Action:** None. |
| IAML0035   | CHANGEDSNT REQUEST SENT TO REMOTE RLSID=rlsid  
**Reason:** A CHANGEDSNT operator command issue to an IAM/PLEX address space on one LPAR is propagated to other IAM/PLEX address spaces running on other LPARs that belong to the same RLSGROUP.  
**Action:** Informational only message. |
| IAML0036   | CHANGEDSNT REQUEST REJECTED FOR RLSID=rlsid  
**Reason:** In response to a CHANGEDSNT operator command, this IAM/PLEX address space attempted to send the request to the IAM/PLEX address space with the indicated RLSID. XCF rejected the send request with a non-zero return code.  
**Action:**  
If the IAM/PLEX address space indicated by the RLSID is either not active or having some type of problem then take appropriate action to correct that error condition.  
If the IAM/PLEX address space indicated appears to be functioning normally, contact Innovation technical support. |
| IAML0037   | REMOTE CHANGEDSNT REQUEST PROCESSED  
**Reason:** The IAM/PLEX address space issuing this message has received and processed a CHANGEDSNT command that was sent from another IAM/PLEX address space.  
**Action:** None, informational only. |
| IAML0047   | GLOBAL TOKEN AREA NOT FOUND FOR THE RLSID PROVIDED  
**Reason:** The IAMRLSX program was executed to delete the Global Token Area from CSA storage, but the Global Token area for the specified IAM RLS address space did not exist. |
IAM/RLS and IAM/PLEX Messages

**Action:** None.

**IAML0048 RLSID MUST BE PROVIDED TO DELETE THE GLOBAL TOKEN AREA**

**Reason:** The IAMRLSX program was executed to delete the Global Token Area from CSA storage, but the user did not provide the RLSID of the Global Token Area to be deleted.

**Action:** Add the required RLSID parameter and rerun.

**IAML0049 ADDRESS SPACE STILL ACTIVE, ISSUE STOP, QUIESCE, OR QUIESCE FORCE TO DEACTIVATE.**

**Reason:** The IAMRLSX program was executed to delete the Global Token Area from CSA storage, but a flag was set in the token area indicating that IAMRLS was still active.

**Action:** This program should not be run until the IAMRLS address space has been deactivated. Quiesce IAMRLS and run this program again.

**IAML0050 GLOBAL NAME TOKEN AREA DELETED**

**Reason:** IAMRLSX successfully deleted the Global Token Area from CSA storage.

**Action:** None

**IAML0051 - IAML0075 IAMOPCOM MESSAGES**

**IAML0051 IAMOPCOM WAITING**

**Reason:** IAMOPCOM, the operator communications task for IAMRLS, is waiting for work.

**Action:** None

**IAML0052 IAMOPCOM SHUTTING DOWN**

**Reason:** IAMOPCOM has ended operator communications and is shutting down.

**Action:** None

**IAML0053 IAMOPCOM MODIFY RECEIVED**

**Reason:** IAMOPCOM has received a modify command.

**Action:** None

**IAML0054 IAMOPCOM RECEIVED INVALID COMMAND**

**Reason:** IAMOPCOM has received an invalid command.

**Action:** Check the spelling and syntax of the command entered.
IAML0055 IAMOPCOM RECEIVED INVALID MODIFY

Reason: IAMOPCOM has received an invalid modify command.

Action: Check the spelling and syntax of the modify command entered.

IAML0056 TRACE COMMAND COULD NOT BE PROCESSED

Reason: IAMOPCOM was unable to load the trace processing program.

Action: Check the contents of the IAM load library to ensure all of the distributed modules are present.

IAML0057 JSWITCH COMMAND COULD NOT BE PROCESSED

Reason: The JSWITCH command was entered to switch the journal datasets but IAMRLS is using the SYSTEM LOGGER for the journal so the switch could not be processed.

Action: None

IAML0058 IAMOPCOM LOAD OF IAMBSTAE FAILED, ESTAE NOT ACTIVE

Reason: IAMOPCOM was unable to load the IAMBSTAE module.

Action: Check the contents of the IAM load library to ensure all of the distributed modules are present.

IAML0059 IAMOPCOM ESTAE ACTIVATION FAILED

Reason: IAMOPCOM was unable to activate its estae routine.

Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0060 IAMOPCOM COMMAND INVALID, MISMATCHED QUOTES

Reason: On an APPLY or RESTORE command, the dataset name entered contains mismatched quotes.

Action: Re-enter the command correctly.

IAML0061 IAMOPCOM COMMAND INVALID, MISMATCHED PARENTHESIS

Reason: On an APPLY or RESTORE command, the dataset name entered contains mismatched parenthesis.

Action: Re-enter the command correctly.

IAML0062 MESSAGES COMMAND INVALID

Reason: An invalid MESSAGES command was entered.

Action: Check the syntax of the command and re-enter it correctly.
**IAML0063**  DISPLAY COMMAND INVALID  
**Reason:** An invalid DISPLAY command was entered.  
**Action:** Check the syntax of the command and re-enter it correctly.

**IAML0064**  SELECT TABLE DATASET LIST  
**Reason:** This is the header message for the list of SELECT TABLE datasets displayed as a result of the DISPLAY,SELECTTB command.  
**Action:** None

**IAML0065**  DSN: dsname or NO MATCHING DATASETS FOUND  
**Reason:** This message is written once for every dataset in the Select Table that matches the selection criteria or once if no datasets are found.  
**Action:** None

**IAML0066**  EXCLUDE TABLE DATASET LIST  
**Reason:** This is the header message for the list of EXCLUDE TABLE datasets displayed as a result of the DISPLAY,EXCLUDETB command.  
**Action:** None

**IAML0067**  DSN: dsname or NO MATCHING DATASETS FOUND  
**Reason:** This message is written once for every dataset in the Exclude Table that matches the selection criteria or once if no datasets are found.  
**Action:** None

**IAML0068**  NO SELECT TABLE DATASET LIST PRESENT  
**Reason:** A DISPLAY,SELECTTB command was entered and no Select Table exists.  
**Action:** None

**IAML0069**  NO EXCLUDE TABLE DATASET LIST PRESENT  
**Reason:** A DISPLAY,EXCLUDETB command was entered and no Exclude Table exists.  
**Action:** None

**IAML0070**  RELEASELOCKS COMMAND INVALID  
**Reason:** An invalid RELEASELOCKS command was entered.  
**Action:** Check the syntax of the command and re-enter it correctly.
IAML0071CLOSEFILE COMMAND INVALID

**Reason:** An invalid CLOSEFILE command was entered.

**Action:** Check the syntax of the command and re-enter it correctly.

IAML0072GETMAIN FAILED ON CLOSEFILE COMMAND

**Reason:** The GETMAIN for an OCQ needed for the CLOSEFILE command failed, the close is not done.

**Action:** Check the region size of the IAMRLS address space.

IAML0073CLOSE COMMAND FAILED, O/C TASK NOT ACTIVE.

**Reason:** The OPEN CLOSE TASK was not active and the close command could not be processed.

**Action:** The QUIESCE,FORCE command may have been entered, if so all files will have been closed. If not then call Innovation for further assistance in diagnosing this problem.

IAML0074RETAINED LOCKS PRESENT, QUIESCE,FORCE REQUIRED TO OVERRIDE

**Reason:** IAMRLS will not shut down while there are record locks that have been retained due to transaction or job step abend that have not yet been recovered.

**Action:** Make sure all recovery action has been processed. If you decide not to have the recovery performed, you can either issue the release retained locks commands, or do the QUIESCE with the FORCE operand.

IAML0075 ECHO OF WHATEVER OPERATOR COMMAND IS ENTERED

**Reason:** This message is issued in confirmation that the IAM/RLS or IAM/PLEX address space has received the indicated operator command.

**Action:** None required, informational only message.

IAML0076EXTRA I/O TASK ATTACHED

**Reason:** An extra I/O subtask was attached to handle an increase in workload.

**Action:** None

IAML0077I/O TASK ATTACHED

**Reason:** This message is issued once for every I/O subtask attached during initialization, till the MINIOTASK is reached.

**Action:** None
**IAM0078** | IAMOPCOM APPLY COMMAND INVALID, NO DSN WAS PROVIDED  
*Reason:* An APPLY command was issued to IAMRLS but no DSN was provided to use as input for command.  
*Action:* Reissue the command with a valid DSN.

**IAM0086** | RLSLOGDD ALLOCATION FAILED, LOGGING DISABLED  
*Reason:* The RLSLOGDD DD statement was not present in the JCL used to start the IAMRLS address space and IAMRLS was not able to dynamically allocate it. Logging is disabled.  
*Action:* If you want logging active, stop IAMRLS and add the RLSLOGDD DD and restart IAMRLS.

**IAM0087** | RLSLOGDD DEALLOCATED BY LOGSWITCH COMMAND  
*Reason:* The RLSLOGDD has been deallocated due to the issuance of the LOGSWITCH command.  
*Action:* None

**IAM0088** | RLSLOGDD REALLOCATED BY LOGSWITCH COMMAND  
*Reason:* The RLSLOGDD has been reallocated due to the issuance of the LOGSWITCH command.  
*Action:* None

**IAM0089** | RLSLOGDD SWITCH COMPLETED  
*Reason:* The switch of the RLSLOGDD to a new sysout dataset has been completed.  
*Action:* None

**IAM0090** | RLSLOGDD DEALLOCATION FAILED, LOGSWITCH TERMINATED  
*Reason:* The RLSLOGDD could not be deallocated, the LOGSWITCH command has been terminated.  
*Action:* The JOURNAL subtask should have printed out the DYNAMIC ALLOCATION messages indicating the reason for the failure.

**IAM0091** | IAMBDSNT PROCESSING COMPLETED  
*Reason:* The processing of the SELECT and EXCLUDE lists is complete.  
*Action:* None

**IAM0092** | RLSLOGDD ALLOCATION FAILED, LOGGING DISABLED  
*Reason:* The RLSLOGDD could not be reallocated, logging to the RLSLOGDD is disabled.
**Action:** Examine the log for messages indicating the reason for the allocation failure. Call Innovation if you cannot correct the error situation.

**IAML0093 RLSLOGDD LOGSWITCH COMPLETE**

**Reason:** The LOGSWITCH command has completed successfully.

**Action:** None

**IAML0094 RLSLOGDD ALLOCATED TO SYSOUT BY IAMLOGER**

**Reason:** The RLSLOGDD has been dynamically allocated during IAMRLS initialization due to the lack of the RLSLOGDD DD statement in the RLS proc.

**Action:** None

**IAML0095 RLSLOGDD DYNALLOC ERROR, RC= ,RSN=**

**Reason:** The dynamic allocation error codes describing the reason for the dynamic allocation failure.

**Action:** Take corrective action if possible otherwise call Innovation if you cannot correct the situation.

**IAML0096 RLSLOGDD ALLOCATED VIA JCL, CANNOT SWITCH**

**Reason:** The LOGSWITCH command cannot switch the RLSLOGDD if it was allocated via a DD statement in the RLS PROC.

**Action:** None

**IAML0100 TRACE DWA GETMAIN FAILURE - TRACE REQUEST FAILED**

**Reason:** The trace processor was unable to obtain storage for its work area.

**Action:** Check the region size of the IAMRLS address space and increase if needed.

**IAML0101 TRACE SUBTASK NOT ACTIVE - TRACE REQUEST FAILED**

**Reason:** The TRACE subtask is not active, tracing cannot be activated.

**Action:** Check the RLSLOGDD and JESLOG to determine why the TRACE subtask was not started. Call Innovation for help in diagnosing this problem.

**IAML0102 TRACE REQUEST COMPLETED SUCCESSFULLY**

**Reason:** A TRACE request has completed successfully.

**Action:** None

**IAML0103 TRACE REQUEST FAILED**

**Reason:** A TRACE request has failed.
Action: Call Innovation for help in diagnosing the problem.

**IAML0104 INVALID TRACE KEYWORD - TRACE REQUEST FAILED**

Reason: An invalid keyword was entered on a TRACE request.

Action: Check the syntax of the command entered and re-enter it correctly.

**IAML0105 TRACE ID REQUIRED. ENTER ID= OR CANCEL**

Reason: A trace ID is required on a TRACE request.

Action: Enter a trace ID or CANCEL in response to the WTOR.

**IAML0106 TRACE JOB AND/OR DSN REQUIRED. ENTER JOB=, DSN= OR CANCEL**

Reason: A jobname and/or a dataset name is required for a TRACE request.

Action: Enter a JOB=jobname or DSN=dsname or CANCEL in response to the WTOR.

**IAML0108 TRACE TYPE REQUIRED. ENTER TR= OR CANCEL**

Reason: The type of TRACE records to record is required on a TRACE request.

Action: Enter one or more of the record types in response to the WTOR, for example: TR=(IOS,IOE,BFR,EXCP,XTND,PC)

**IAML0109 TRACE REQUEST MUST BE START, STOP, LIST OR CANCEL**

Reason: One of the above keywords must be present in the TRACE request.

Action: Re-enter the TRACE request specifying either START, STOP, LIST, or CANCEL.

**IAML0110 TRACE_ID=xxxxxxxx**

Reason: Output from a TRACE request showing the TRACE ID.

Action: None

**IAML0111 TYPE=IOS,IOE,BFR,EXCP,XTND,PC**

Reason: Output from a TRACE request showing the TRACE TYPES.

Action: None

**IAML0112 DSN=dsname**

Reason: Output from a TRACE request showing the DSNNAME being traced.

Action: None
IAML0113  JOBNAME=jobname

Reason: Output from a TRACE request showing the JOBNAME being traced.
Action: None

IAML0114  STEPNAME=stepname

Reason: Output from a TRACE request showing the STEPNAME.
Action: None

IAML0115  OUTDD=ddname

Reason: Output from a TRACE request showing the OUTPUT DD.
Action: None

IAML0116  TRACE_ID ALREADY ACTIVE. TRACE REQUEST FAILED

Reason: A TRACE request with the same TRACE ID was already active, the current request is terminated.
Action: If this is a new request use a different TRACE ID, if you want to change the old request, STOP it first then restart it with the new options.

IAML0117  SPECIFIED TRACE_ID NOT FOUND. TRACE NOT DEACTIVATED

Reason: A TRACE,STOP command was issued but an active TRACE with the ID requested was not active.
Action: None

IAML0118  NO ACTIVE TRACE REQUESTS FOUND

Reason: No active trace requests were found to list on the TRACE,LIST command.
Action: None

IAML0119  TRE GETMAIN FAILURE. TRACE REQUEST FAILED

Reason: A GETMAIN for a trace element failed causing the trace request to be failed.
Action: Check the region size of the IAMRLS address space, contact Innovation for assistance in diagnosing this problem.

IAML0120  TRACE OUTPUT OPEN FAILURE. TRACE REQUEST FAILED

Reason: The TRACE processor could not open the requested trace output file.
Action: Make sure the file is allocated and has sufficient space.
IAML0121  TRACE OUTPUT DCB GETMAIN FAILURE. TRACE REQUEST FAILED
Reason: The GETMAIN for the trace output DCB failed. The TRACE request is failed.
Action: Check the region size of the IAMRLS address space, contact Innovation for assistance in diagnosing this problem.

IAML0122  OUTPUT DD FOR TRACE_ID xxxxxxxx IS ddname
Reason: Output from a TRACE request showing the OUTPUT DD.
Action: None

IAML0123  IAM ADDRESS SPACE TRACE REQUEST:
Reason: Output header line for a trace request.
Action: None

IAML0124  TRACE_ID xxxxxxxx DEACTIVATED
Reason: The trace id in the message has been deactivated.
Action: None

IAML0125  TRACE_ID xxxxxxxx QUEUED
Reason: A trace request has been queued and will become active when the requested selection criteria have been met.
Action: None

IAML0126  TRACE_ID xxxxxxxx ALREADY ACTIVE
Reason: An active trace request is already active with the trace id requested on a new trace request.
Action: If this is indeed a new request, choose a new trace id.

IAML0127  INVALID TRACE KEYWORD IN PARMLIB - TRACE REQUEST FAILED
Reason: An invalid trace keyword was detected, the trace request fails.
Action: Correct the trace keyword in error.

IAML0128  INVALID DSN SPECIFIED - TRACE REQUEST FAILED
Reason: An invalid dataset name was specified on a trace request, the trace request is failed.
Action: Re-enter the trace request with a valid dataset name.
IAML0129 DATASET ALLOCATION FAILED

Reason: Allocation of the output dataset for a trace request failed.

Action: Make sure the output dataset exists and is available for use.

IAML0130 LOG OF USER TRACE REQUEST

Reason: Log entry of the trace request as entered by the user.

Action: None

IAML0150 dsname OPENED DD=ddname J=jobname S=stepname I=jobid

Reason: DSNAME X.Y.Z DDNAME DDDDDDDDD under JOBNAME JJJJJJJ and STEPNAME SSSSSSSS using PROGRAM PPPPPPPP was successfully opened by IAMRLS.

Action: None

IAML0151 dsname OPEN FAILED DD=ddname J=jobname S=stepname I=jobid

Reason: The indicated dataset was not successfully opened by IAMRLS.

Action: None

IAML0152 dsname DYNALOC FAILED RC=nnnn DD=ddname J=jobname S=stepname I=jobid

Reason: The indicated dataset could not be dynamically allocated by IAMRLS.

Action: None

IAML0153 dsname CLOSED DD=ddname J=jobname S=stepname I=jobid

Reason: The indicated dataset was successfully closed by IAMRLS.

Action: None

IAML0154 dsname CLOSE FAILED DD=ddname J=jobname S=stepname I=jobid

Reason: DSNAME X.Y.Z DDNAME DDDDDDDDD under JOBNAME JJJJJJJ and STEPNAME SSSSSS using PROGRAM PPPPPPPP was not successfully closed by IAMRLS.

Action: None

IAML0155 dsname DYNDAL LOC FAILED RC=nnnnn DD=ddname J=jobname

Reason: IAMRLS attempted to de-allocate the indicated dataset, but encountered a failure.

Action: Determine cause of the problem from the return codes. Contact Innovation technical support if further assistance is needed.
IAML0156  dsname CLOSE COMMANDS ISSUED DUE TO IAMRLS REQUEST

Reason: In response to an operator command the indicated dataset has been closed.

Action: None

IAML0157  dsname OPEN  REQUESTED BY RLSID=rlsid CPUID=xxxxxxxx

Reason: The indicated dataset has been opened for processing by the indicated IAMPLEX region.

Action: None.

IAML0158  dsname CLOSE REQUESTED BY RLSID=rlsid CPUID=xxxxxxxx

Reason: The indicated dataset has been closed due to a request from the indicated IAMPLEX region.

Action: None

IAML0167  IAMBROC-IAMBSTAE LOAD FAILED - ESTAE NOT ACTIVE

Reason: The remote open/close task that is responsible for sending those requests to a remote IAM/PLEX instance encountered a non-zero return code when attempting to load the IAMBSTAE load module. The task will continue to operate however will not have ESATE protection for recovery should the task experience an abend condition.

Action: Notify Innovation technical support for assistance.

IAML0168  IAMBROC-ESTAE ACTIVATION FAILED

Reason: The remote open/close task that is responsible for sending those requests to a remote IAM/PLEX instance encountered a non-zero return code when attempting to activate an ESTAE environment. The task will continue to operate however will not have ESATE protection for recovery should the task experience an abend condition.

Action: Notify Innovation technical support for assistance.

IAML0169  IAMBROC-SCHEDULE FAILED RC=xxxxxxxx J=jobname S=stepname I=jobid

Reason: The request to schedule an SRB in the address space identified by the indicated job step failed with the indicated return code, displayed as hex digits. This SRB was intended to resume processing within that job step on completion of the open or close request being handled by IAM/PLEX.

Action: If the indicated job step is no longer active, then no action is necessary. If the indicated job step is active, obtain a dump of this IAM/PLEX address space and the batch job indicated then notify Innovation technical support for assistance. The job will most likely have to be cancelled and restarted.
IAML0170  IAMBQP - PAUSE/RELEASE ERROR - XXXXXXXX - RC=XXXXXXXX

**Reason:** An I/O subtask received an error during PAUSE or RELEASE processing.

**Action:** None

IAML0173  RLSID=rlsid IS NO LONGER CONNECTED TO GROUP=rlsgroup

**Reason:** The IAM/PLEX instance identified by the indicated RLSID is no longer connected to the indicated RLSGROUP. Message is issued during termination processing of the indicated IAM/PLEX instance.

**Action:** None.

IAML0174  REMOTE NAME TOKEN AREA DELETED FOR RLSID=NNNN

**Reason:** The IAM/PLEX instance identified by the indicated RLSID has deleted its name token area. Message is issued during termination processing of the indicated IAM/PLEX instance.

**Action:** None.

IAML0175  RLS SYSPLEX GROUP INFORMATION FOR RLSID=CCCC

**Reason:** This message appears in response to a DISPLAY,GROUP command issued on an IAM/PLEX instance identified by the indicated RLSID. It will be followed by subsequent IAML0176 messages.

**Action:** None.

IAML0176  RLSID=rlsid,CPUID=cccccccc,STATUS=[ACTIVE|CREATED|QUIESCED|FAILED|NOT-DEF]

**Reason:** This message appears in response to a DISPLAY,GROUP command issued on an IAM/PLEX instance identified by the indicated RLSID on the preceding IAML0175 message. This message displays the known IAM/PLEX instances by RLSID and CPUID that are associated with the same RLSGROUP indicated by the IAML0175 message, along with the current status.

**Action:** None.

IAML0177  REMOTE OPEN REQUESTED RLSID=rlsid CPUID=cccccccc

**Reason:** An IAM/PLEX address space received a request to open an IAM file that is being processed by the remote RLS instance identified by the RLSID on the message. This IAM/PLEX instance issuing this message is acting as a ROUTER in this circumstance, and will send the open request to the indicated TARGET IAM/PLEX instance.

**Action:** None, informational message. This message is only displayed when the MESSAGES=ALL parameter has been specified.
IAML0178  REMOTE OPEN COMPLETED RLSID=rlsid CPUID=cccccccc RC=xxxxxxxx

Reason: An IAM/PLEX address space received a request to open an IAM file that is being processed by the remote RLS instance identified by the RLSID on the message. This IAM/PLEX instance issuing this message is acting as a ROUTER in this circumstance having received a response from the TARGET RLSID with the associated return code has completed the open processing.

Action: None, informational message. This message is only displayed when the MESSAGES=ALL parameter has been specified.

IAML0179  REMOTE CLOSE REQUESTED RLSID=rlsid CPUID=cccccccc

Reason: An IAM/PLEX address space received a request to close an IAM file that has opened by the remote RLS instance identified by the RLSID on the message. This IAM/PLEX instance issuing this message is acting as a ROUTER in this circumstance, and will send the close request to the indicated TARGET IAM/PLEX instance.

Action: None, informational message. This message is only displayed when the MESSAGES=ALL parameter has been specified.

IAML0180  REMOTE CLOSE COMPLETED RLSID=rlsid CPUID=cccccccc RC=xxxxxxxx

Reason: An IAM/PLEX address space received a request to close an IAM file that has been opened by the remote RLS instance identified by the RLSID on the message. This IAM/PLEX instance issuing this message is acting as a ROUTER in this circumstance having received a response from the TARGET RLSID with the associated return code has completed the close processing.

Action: None, informational message. This message is only displayed when the MESSAGES=ALL parameter has been specified.

IAML0181  REMOTE TERMINATE REQUESTED RLSID=rlsid CPUID=cccccccc

Reason: An IAM/PLEX address space received a request for job step termination of a job step that had one or more IAM files handled by the remote RLS instance identified by the RLSID on the message. This IAM/PLEX instance issuing this message is acting as a ROUTER in this circumstance and has sent the job step termination request to the TARGET RLSID.

Action: None, informational message. This message is only displayed when the MESSAGES=ALL parameter has been specified.

IAML0182  REMOTE TERMINATE COMPLETEx RLSID=rlsid CPUID=cccccccc RC=xxxxxxxx

Reason: An IAM/PLEX address space received a request for job step termination of a job step that had one or more IAM files handled by the remote RLS instance identified by the RLSID on the message. This IAM/PLEX instance issuing this message is acting as a ROUTER in this circumstance having received a response from the TARGET RLSID has completed termination processing.

Action: None, informational message. This message is only displayed when the MESSAGES=ALL parameter has been specified.
IAML0183  IAMBROC RESPONSE RECEIVED FROM TARGET RLSID=rlsid CPUID=cccccccc

**Reason:** An IAM/PLEX remote open/close task has received a response from a prior request to a target address space.

**Action:** None, informational message. This message is only displayed when the MESSAGES=ALL parameter has been specified.

IAML0184  IAMBROC XCF FAILURE DURING cccc REQUEST/RESPONSE, TARGET RLSID=rlsid

**Reason:** An error was encountered either sending or receiving a message from XCF. There should be a message preceding this message with more details.

**Action:** Search for preceding message for details on the failure, and take the associated action.

IAML0185  IAMBROC function IXCMSGO FAILURE RC=xx REAS=xxxxxxxx TARGET=rlsid CPUID=cupid

**Reason:** The transmission of the indicated request type that was destined for the indicated target IAM/PLEX address space failed with the indicated return and reason code.

**Action:** Check the return code and reason code to determine the cause of the error. If it is an environmental factor, correct the problem. Take a dump of the IAM/PLEX address space that issued the message with the IAMPLEX operator command of DUMP. Then contact Innovation technical support for assistance if needed.

IAML0186  IAMBROC TERM RLSID=rlsid NOT CURRENTLY ACTIVE

**Reason:** A router IAM/PLEX instance received a job step termination request which needed to be sent to the indicated TARGET IAM/PLEX instance, however the Target IAM/PLEX address space was not active.

**Action:** No action is necessary if the indicated IAM/PLEX region is not active. If it is active, enter a DISPLAY,GROUP command and dump the IAM/PLEX address space, then contact Innovation technical support with the requested information.

IAML0187  IAMRLS ROUTER ENTRY NOT FOUND AFTER function RLSID=rlsid...

**Reason:** The Target IAM/PLEX did not find the indicated Router IAM/PLEX instance. Because of that, it was unable to return a response back to the originating system.

**Action:** If the Target system is still running and the job did not abend, the job will need to be cancelled.

IAML0188  IAMBOC function XCF RESPONSE SENT TO ROUTER

**Reason:** The Target IAM/PLEX instance has responded to the Router IAM/PLEX upon completion of the indicated function.

**Action:** None, informational only message.
**IAML0189 IAMBOC EXTERNAL function status**

**RLSID=rlsid CPUID=cupid MSGID=msgid JOB=jobname**

**Reason:** This message is displayed by the TARGET IAM/PLEX address space indicating the processing of an open or close request from an external ROUTER IAM/PLEX instance.

**Action:** None, this is an informational message that is only displayed when the MESSAGES=ALL parameter is set.

**IAML0190 dfsname OPEN-type DD=ddname J=jobname S=stepname I=job####**

**Reason:** Message issued by IAM/PLEX ROUTER on completion of an open request on a TARGET system. The "type" will be either "I" for input only, or "U" for update. The rest of the information provided is based on the job that had requested the open.

**Action:** None, informational only message followed by an IAML0191 message.

**IAML0191 dfsname OPENED ON RLSID=rlsid CPUID=xxxxxxxx**

**Reason:** Message issued by IAM/PLEX ROUTER on completion of an open request on a TARGET system. This message will follow the IAML0190 message, and provide information on the TARGET IAM/PLEX instance that actually did the open.

**Action:** None, informational only message followed by an IAML0191 message.

**IAML0192 dfsname CLOSED DD=ddname J=jobname S=stepname I=job####**

**Reason:** Message issued by IAM/PLEX ROUTER on completion of a close request on a TARGET system. The rest of the information provided is based on the job that had requested the open.

**Action:** None, informational only message followed by an IAML0191 message.

**IAML0193 Dsname CLOSED ON RLSID=rlsid CPUID=xxxxxxxx**

**Reason:** Message issued by IAM/PLEX ROUTER on completion of a close request on a TARGET system. This message will follow the IAML0192 message, and provide information on the TARGET IAM/PLEX instance that actually did the close.

**Action:** None, informational only message followed by an IAML0191 message.

**IAML0194 IAMBPC06 OCQ REQUEUED DUE TO TARGET FAILURE**

**Reason:** The target RLS is not available.

**Action:** The requested close will not occur on the target IAM PLEX region. The requesting job / address space will have the file indicated as being closed.
IASML0195  RLSID=XXXXX CONNECTED TO GROUP  NNNNNNNN

**Reason:** The XCF exit has determined that this RLSID for an IAM PLEX address space is connected to the specified group.

**Action:** None.

IASML0196  dataset name NO OID  DD=ddname  J=jobname  S=stepname  I=jobid

**Reason:** Record lock manager was unable to create an owner identifier block.

**Action:** Contact Innovation for support

IASML0200  INSUFFICIENT STORAGE FOR ADDITIONAL RECORD LOCKS

**Reason:** IAMBRLOK was unable to obtain more storage for lock tables. The I/O that requested the record lock is failed.

**Action:** Make sure IAMRLS has sufficient storage in the area being used for the record locks.

IASML0201  INSUFFICIENT STORAGE FOR ADDITIONAL OWNER ELEMENTS

**Reason:** IAMBRLOK was unable to obtain more storage for owner elements. The I/O that requested the record lock is failed.

**Action:** Make sure IAMRLS has sufficient region size in the startup jcl.

IASML0202  INSUFFICIENT STORAGE FOR DEADLOCK DETECTION
INSUFFICIENT SERIALIZATION WORK AREAS

**Reason:** IAMBRLOK was unable to obtain more storage for the indicated area. For deadlock detection, processing continues, however a deadlock situation is possible. For other reasons, request is failed with a logical error of x’28’

**Action:** Make sure IAMRLS has sufficient region size in the startup jcl. Contact Innovation for support after dumping the region.

IASML0203  RECORD LEVEL LOCKING ALREADY INITIALIZED

**Reason:** A call has been made to IAMBRLOK to perform initialization and IAMBRLOK has detected that initialization has already been done.

**Action:** Call Innovation for help in diagnosing this problem.

IASML0204  INSUFFICIENT STORAGE FOR LOCK HASH TABLE

**Reason:** IAMBRLOK was unable to obtain more storage for a lock hash table.

**Action:** Make sure IAMRLS has sufficient region size in the startup jcl.
IAML0205  INSUFFICIENT STORAGE FOR RECORD LOCKS

Reason: IAMBRLOK was unable to obtain storage for record locks during initialization. The startup of IAM/PLEX instance will be failed.

Action: Either increase storage via REGION or if locks are to be in 64-bit storage increase the MEMLIMIT parameter for more storage. Alternatively, reduce the value for MAXLCKS. Then restart the failing IAM/PLEX instance.

IAML0206  INSUFFICIENT STORAGE FOR LOCK OWNER TABLE

Reason: IAMBRLOK was unable to obtain more storage for lock owner tables. The IAM/PLEX or IAM/RLS startup has failed.

Action: Make sure IAMRLS has sufficient region size in the startup jcl.

IAML0207  INSUFFICIENT STORAGE FOR LOCK OWNER POOL

Reason: IAMBRLOK was unable to obtain storage for a lock owner pool. The IAM/PLEX or IAM/RLS startup has failed.

Action: Make sure IAMRLS has sufficient region size in the startup jcl.

IAML0208  INSUFFICIENT STORAGE FOR DEADLOCK DETECTION

Reason: IAMBRLOK was unable to obtain storage for deadlock detection tables. The IAM/PLEX or IAM/RLS startup has failed.

Action: Make sure IAMRLS has sufficient region size in the startup jcl.

IAML0209  INSUFFICIENT STORAGE FOR LOCK WORK AREA

Reason: IAMBRLOK was unable to obtain storage for a lock work area. If this occurs during initialization of the IAM/RLS or IAM/PLEX instance, the startup will be failed. If this occurs during regular processing, the request being process will be failed.

Action: Make sure IAMRLS has sufficient region size in the startup jcl.

IAML0210  DEADLOCK CONDITION DETECTED, I/O REQUEST FAILED FOR J=jobname S=stepname D=ddname

Reason: A deadlock condition was detected for DD=ddname J=JOBNAME S=STEPNAME. The I/O request is failed.

Action: Wait for the other user of the resource to end and resubmit the job.

IAML0211  LOCKS HAVE BEEN RETAINED FOR JOB ASID=xxxx J=jobname S=stepname I=jobid

Reason: In response to a DISPLAY,RETAINEDLOCKS, this message indicates for what job steps record locks were retained due to an abend.

Action: None, informational message only.
IAM/RLS and IAM/PLEX Messages

IAML0212  RL DSN=dsname  ASID=xxxx  J=jobid  S=stepname  I=jobid

Reason: In response to a DISPLAY, RETAINEDLOCKS, DSN. This message indicates the dataset name, and associated jobstep for which record locks have been retained. A group of these messages is preceded by message IAML0211.

Action: None, informational message only.

IAML0213  NO LOCKS RETAINED

Reason: In response to a DISPLAY, RETAINEDLOCKS command, this message indicates that there are no retained locks.

Action: None, informational message only.

IAML0214  RETAINED LOCKS FOUND, SEE RLSLOGDD

Reason: In response to a DISPLAY, RETAINEDLOCKS command. The list of retained locks is displayed in the RLSLOGDD.

Action: None, informational message only.

IAML0215  Dsname LOCKED  ASID=xxxx  J=jobname  S=stepname  I=jobid

Reason: In response to a DISPLAY, CONTENTION command, indicates that the specified job has record locks held for the identified dataset.

Action: None, informational message only.

IAML0216  MAXLOCK CONDITION DETECTED, I/O REQUEST FAILED FOR  DD=12345678  J=12345678  S=12345678

Reason: The batch job indicated in the message attempted to exceed concurrently holding more record locks than specified by the IAM/RLS or IAM/PLEX startup parameter value for MAXLOCKS. This results in failing this and any subsequent I/O requests that require record locks unless locks are released. The reason for this action is to prevent IAM/RLS or IAM/PLEX address space failures due to storage shortages that can be caused by holding an excessive number of record locks at one time.

Action: This batch job step needs to be revised to either (1) use the AUTOSYNCPOINT function, (2) incorporate calls to the IAM batch Syncpoint routine to release locks, or (3) set some of the files being processed to a non-recoverable state by turning off before image journaling.

IAML0217  NO RECORD LOCK CONTENTION FOUND

Reason: In response to a DISPLAY, CONTENTION command, indicates that no record lock contention was detected, meaning that no I/O requests are waiting for record lock.

Action: None, informational message only.
IAM/RLS and IAM/PLEX Messages 80.05

IAML0218 RECORD LOCK CONTENTION FOUND, SEE RLSLOGDD

**Reason:** In response to a DISPLAY,CONTENTION command, indicates that record lock contention exists. This means that some I/O requests are waiting for a record lock. Additional details on what is waiting can be found in messages in the RLSLOGDD.

**Action:** Review the RLSLOGDD for more information on the lock contention.

IAML0219 LOCK CONTENTION FOUND FOR FOLLOWING KEY:

**Reason:** In response to a DISPLAY,CONTENTION command, indicates the specified records that are experiencing contention.

**Action:** None, informational message only.

IAML0220 UNABLE TO OBTAIN 64-BIT STORAGE FOR LOCKS RC= xx REASON=xxxxxxxx REQUESTED x,xxx,xxx MEGABYTES OF 64-BIT STORAGE

**Reason:** During initialization of an IAM/PLEX or IAM/RLS address space, IAM was not able to obtain the required amount of 64-bit virtual storage to use for the record locking structures. The IAM/PLEX or IAM/RLS address space will terminate due to this problem.

**Action:** Make sure to provide an adequate MEMLIMIT value on the EXEC card of the PROC or in the system SMFPRMxx so that IAM can obtain the necessary storage. Alternatively the amount of storage requested can be reduced by lowering the MAXLOCKS parameter however that may result in subsequent failures if the number of concurrently held record locks requires more storage and is unable to obtain it at that time.

IAML0222 BRLOK SYNCPOINT

**Reason:** A SYNCPOINT request was issued by the indicated job or CICS transaction.

**Action:** None, this message is informational only and will only be displayed when the MESSAGES=ALL parameter has been set.

IAML0223 IAMBRLOK ENTERED FOR CICS SYNCPOINT/TX END

**Reason:** A SYNCPOINT request was issued by a CICS transaction.

**Action:** None, this message is informational only and will only be displayed when the MESSAGES=ALL parameter has been set.

IAML0224 IAMBRLOK FOUND MATCHING CICS TX OID

**Reason:** Indicates that IAMBRLOK found an entry for the CICS transaction that had issued a SYNCPOINT and will perform the syncpoint processing.

**Action:** None, this message is informational only and will only be displayed when the MESSAGES=ALL parameter has been set.
IAM/RSL and IAM/PLEX Messages

IAML0225 IAMBRLOK ENTERED FOR JOB STEP TERMINATION

Reason: IAMBRLOK was entered to perform cleanup of locks for a job step termination.

Action: None, this message is informational only and will only be displayed when the MESSAGES=ALL parameter has been set.

IAML0226 IAMBRLOK JOB STEP END RELEASING LOCKS

Reason: IAMBRLOK has released record locks during a job step end.

Action: None, this message is informational only and will only be displayed when the MESSAGES=ALL parameter has been set.

IAML0227 IAMBRLOK JOB ABENDED, RETAINING LOCKS

Reason: IAMBRLOK has retained record locks after a job abend.

Action: Insure that a backout recovery is performed for this file, then release the locks if not already done so by IAMRLS.

IAML0228 IAMBRLOK BATCH SYNCPOINT RELEASING LOCKS

Reason: IAMBRLOK has been called from the batch syncpoint exit to release locks following a syncpoint exit call from the batch program.

Action: None, this message is informational only and will only be displayed when the MESSAGES=ALL parameter has been set.

IAML0229 IAMBRLOK ENTERED FOR BATCH SYNCPOINT

Reason: IAM RLS lock manager has been called to release locks from a job issuing batch syncpoints.

Action: None, this message is informational only and will only be displayed when the MESSAGES=ALL parameter has been set.

IAML0230 JOBSTEP NORMAL TERMINATION ASID=xxxx J=jobname S=stepname I=jobid

Reason: The indicated job step, which had IAM files opened to IAM RLS, has completed without an abend.

Action: None, informational message only.

IAML0230 JOBSTEP ABEND TERMINATION ASID=xxxx J=jobname S=stepname I=jobid

Reason: The indicated job step has abnormally terminated. If the job was accessing recoverable datasets, then any record locks that were held will be retained until a recovery is performed.

Action: None, informational message only.
IAM/RLS and IAM/PLEX Messages

IAML0230  ADDRSSPACE ABEND TERMINATION ASID=xxxx J=jobname S=stepname I=jobid

Reason:  IAMBTERM has detected an address space termination of a job that was processing IAM datasets under IAM RLS. This message will only appear if an address space fails without having gone through the step termination.

Action:  None, informational message only.

IAML0232  RECORD LOCKS RETAINED ASID=xxxx J=jobname S=stepname I=jobid

Reason:  Record locks held by the specified job were retained due to an abend or because of address space failure.

Action:  Make sure that a file recovery is performed, which will release the retained locks.

IAML0233  QUEUED DYNAMIC BACKOUT REQUEST ASID=xxxx J=jobname S=stepname I=jobid

Reason:  IAM RLS posted the dynamic job backout task to perform a backout on the indicated job step due to an abend or address space failure.

Action:  Verify from subsequent messages that the backout was performed and that it released the retained locks.

IAML0234  TERMINATION PROCESSING COMPLETED ASID=xxxx J=jobname S=stepname I=jobid

Reason:  IAM RLS step termination processing has completed for the indicated job step. This means that any record locks held by the job step were either released, or retained as needed.

Action:  None, this is an informational message only.

IAML0240  RELEASE RETAINED LOCKS REQUESTED ASID=xxxx J=jobname S=stepname I=jobid

Reason:  A RELEASELOCKS command has been issued for the indicated job. Any information not provided on the command will be printed as asterisks.

Action:  None, informational message only.

IAML0241  FOUND FOLLOWING MATCHING JOB ASID=xxxx J=jobname S=stepname I=jobid

Reason:  In response to a RELEASELOCKS command, IAM RLS has found the indicated job with record locks.

Action:  None, informational message only.

IAML0242  RETAINED LOCKS RELEASED FOR AS ID=xxxx J=jobname S=stepname I=jobid

Reason:  IAM RLS released the retained record locks held for the indicated job step.

Action:  None, informational message only.
IAML0243  REQUESTED JOB NOT FOUND ASID=xxxx J=jobname S=stepname I=jobid

**Reason:** The job specified on a RELEASELOCKS command could not be found.

**Action:** Issue either a DISPLAY,RETAINEDLOCKS or DISPLAY,CONTENTION to get a list of job(s) which currently have locks, and supply more job identification on the RELEASELOCKS command.

IAML0244  NONE RELEASED, MORE THAN 1 MATCH ASID=xxxx J=jobname S=stepname I=jobid

**Reason:** In response to a RELEASELOCKS command, IAM RLS found more than one matching job. No record locks were released.

**Action:** Reissue the RELEASELOCKS command with more specific job identification, as can be found on the preceding IAML0241 messages.

IAML0245  LOCK RELEASE REQUEST FAILED

**Reason:** IAM was unable to release any locks. The job may not have had any retained locks.

**Action:** None.

IAML0246  TERMINATING JOB OID NOT FOUND ASID=xxxx J=jobname S=stepname I=jobid

**Reason:** A job termination request was given to the IAM/RLS or IAM/PLEX address however IAM had no record of the job.

**Action:** None.

IAML0247  IXCMMSGO FAILED, RC=xx REAS=xxxxxxxx RLS=rlsid J=jobname S=stepname I=jobid

**Reason:** A TARGET IAM/PLEX instance processed termination for a job, however the attempt to respond to the ROUTER IAM/PLEX instance failed with the indicated return and reason codes.

**Action:** If unable to correct the problem from the provided information, contact Innovation technical support for assistance.

IAML0250  IAMBREST INITIALIZATION FAILURE. BATCH BACKOUT NOT AVAILABLE

**Reason:** The automatic batch backout was either unable to obtain storage or encountered an invalid environment during initialization and is unable to process.

**Action:** Contact Innovation technical support for assistance.

IAML0251  BACKOUT REQUESTED ASID=xxxx J=xxxxxxxx S=xxxxxxxx I=xxxxxxxx

**Reason:** The automated job backout process has started processing a backout for the failed job that is indicated in the message.

**Action:** None.
IAM0252 BACKOUT COMPLETE ASID=xxxx J=xxxxxxxx S=xxxxxxxx I=xxxxxxxx

Reason: The automated job backout has completed processing for the indicated failed job.

Action: None.

IAM0253 Name_of_data_set_allocated_for BACKOUT DD=xxxxxxxx J=xxxxxxxx I=xxxxxxxx

Reason: Indicates the name(s) of the IAM datasets allocated to updated by the backout process.

Action: None.

IAM0254 J=xxxxxxxx S=xxxxxxxx I=xxxxxxxx (JOURNAL I/O TRACE MSG)

IAM0255 IAMBREST DATASPACE INIT FAILURE. BACKOUT WITHOUT DATASPACE

Reason: IAMBREST requested Data Space storage, however that request failed. IAMBREST continues processing without the use of Data Space storage.

Action: Notify Innovation technical support of problem.

IAM0256 IAMBREST REQUIRED JOURNAL DATASET ALLOCATION FAILURE. BACKOUT NOT PERFORMED

Reason: IAMBREST could not allocated a required journal dataset.

Action: Make sure all journal datasets are properly cataloged and available. If so, then contact Innovation technical support for assistance.

IAM0257 BACKOUT FAILED ASID=xxxx J=jobname S=stepname I=jobid

Reason: The dynamic job backout processing has failed.

Action: Contact Innovation technical support. Attempt to perform backout manually with the IAMJREST utility.

IAM0258 BACKOUT COMPLETE - JOURNAL RECORDS USED - XXXXXXXX ASID=XXXX

Reason: Dynamic Job Backout has performed a backout using the indicated number of journal records.

Action: None.

IAM0259 IAMBREST CAN NOT LOCATE CURRENT JOURNAL - BACKOUT NOT PERFORMED

Reason: Current journal dataset can not be determined.
Action: Restore ends with RC=8

IAML0262  IAMBREST CANNOT POSITION TO STARTING JOURNAL FOR BACKOUT - BACKOUT NOT PERFORME

Reason: Cannot determine staring journal based on timestamps for journal activation and job/step start.

Action: Restore ends with RC=8.

IAML0263  IAMBREST INTERNAL DSN ENTRY NOT FOUND - BACKOUT NOT PERFORMED

Reason: Current journal DSN not found in list of available journals for this execution of IAMRLS

Action: Restore ends with RC=8.

IAML0264  IAMBREST COULD NOT DETERMINE DSN FOR ALLOCATION: dataset.name

Reason: Invalid DSN was passed for Dynamic Allocation.

Action: Restore ends with RC=8.

IAML0265  IAMBREST ALLOCATION FAILED FOR DSN: dataset.name RC=xxxxxx REAS=xx

Reason: Dynamic Allocation Error.

Action: See Return Code (RC=) and Reason Code (REAS=) and take appropriate action.

IAML0266  DJB TRACE ACTIVE - DDNAME=ddname

Reason: Dynamic Job backout trace is active for DDNAME=ddname

Action: None

IAML0267  DJB TRACE ACTIVATION FAILED

Reason: Dynamic Job Backout trace activation failed.

Action: None

IAML0275  IAMRLS STORAGE SHORTAGE OBTAINING IOQ OR OCQ CELL

Reason: IAM/PLEX or IAM/RLS was unable to obtain additional storage for the control block used to manage an I/O or open/close request.

Action: Attempt to take a dump of the IAM/PLEX or IAM/RLS address space with the F iamplex,DUMP operator command, then notify Innovation technical support for assistance.
IAML0276  I/O REQUEST SENT TO   TARGET=rlsid CPUID=cupid MSGID=xxxxxxxx

Reason: The IAM/PLEX instance acting as a router has sent an I/O request to the target IAM/PLEX.

Action: None, informational only. Message will only appear when the MESSAGES=ALL parameter has been specified.

IAML0277  I/O RESPONSE RECEIVED FROM TARGET=xxxx CPUID=xxxxxxxx MSGID=xxxxxxxx

Reason: The IAM/PLEX router has received a response for an I/O request from the target IAM/PLEX.

Action: None, informational only. Message will only appear when the MESSAGES=ALL parameter has been specified.

IAML0278  I/O ERROR RESPONSE FROM TARGET=rlsid CPUID=cupid MSGID=xxxxxxxx

Reason: The IAM/PLEX router has received a response for an I/O request from the target IAM/PLEX that indicates an error was encountered with IAM/PLEX processing.

Action: Obtain a dump of the IAM/PLEX target and router address spaces, and notify Innovation technical support.

IAML0279  I/O ERROR - TARGET TERMINATED - TARGET=rlsid CPUID=cupid

Reason: A router IAM/PLEX address space had been posted by recovery for an I/O request sent to the indicated target.

Action: If the indicated target IAM/PLEX instance is operational, notify Innovation technical support.

IAML0280  I/O EXPECTED RESPONSE - TARGET=rlsid CPUID=cupid

Reason: A router IAM/PLEX address space received an invalid response from the target IAM/PLEX address space for the I/O request.

Action: Take a dump of both the router and target IAM/PLEX address spaces as soon as possible, and contact Innovation technical support for assistance.

IAML0281  IAMBI0 REQ xxxxx xxxx/xxxxxxxx MSGID=xxxxxxxx LEN=xxxxxxxx

Reason: The target IAM/PLEX address space has received the indicated I/O request from the router IAM/PLEX address space and is beginning processing of that request.

Action: None, informational only message that is only issued when the MESSAGES=ALL parameter is set.
IAML0282  IAM BIO RESP xxxX xxx/xxxxxxxx MSGID=xxxxxxxx FDBK=xxxxxxxx

Reason: The target IAM/PLEX address space has completed processing the indicated I/O request and is sending the response back to the router.

Action: None, informational only message that is only issued when the MESSAGES=ALL parameter is set.

IAML0283  IAMBPC07 IOQ=xxxxxxxx OBTAINED FOR REMOTE I/O OCQ=xxxxxxxx

Reason: The router IAM/PLEX has received an I/O request destined for another IAM/PLEX address space, and has obtained the internal control block necessary for the request.

Action: None, informational only message that is only issued when the MESSAGES=ALL parameter is set.

IAML0284  IAMBPC07 IOQ=xxxxxxxx REQUEUED AFTER REMOTE I/O OCQ=xxxxxxxx

Reason: The router IAM/PLEX has received a response from the target IAM/PLEX address space for an I/O request and is releasing the internal control block necessary for the request.

Action: None, informational only message that is only issued when the MESSAGES=ALL parameter is set.

IAML0285  IAMBPC02 SYNCPOINT REQUEST IOQ=xxxxxxxx

Reason: The router IAM/PLEX has received a syncpoint request.

Action: None, informational only message that is only issued when the MESSAGES=ALL parameter is set.

IAML0286  IAMBPC03 TRACE MESSAGES

Reason: Trace messages issued from IAMBPC03 when MESSAGES=ALL is specified and IAMBMON is in use.

Action: None.

IAML0287  IAMBPC02 SYNCPOINT Release Fail

Reason: Release of task to perform SYNCPOINT request failed.

Action: Contact Innovation for support.

IAML0300  RECORD LOCK RECOVERY INITIALIZING

Reason: Message occurs during startup of an IAMRLS or IAMPLEX address space to indicate that the Record Lock Recovery function is being initialized. Subsequent messages will be provided on the initialization process and the status of the Record Lock Recovery when the initialization has been completed.

Action: None. This is an informational message.
IAML0301 RECORD LOCK RECOVERY ACTIVE, CHECKPOINT EVERY XXX MINUTES

**Reason:** The message indicates that the Record Lock Recovery function has successfully been initialized and is currently active. The checkpoint frequency is indicated.

**Action:** None. This is an informational message.

IAML0302 INSUFFICIENT STORAGE LOCK RECOVERY WORK AREA

**Reason:** The Record Lock Recovery function was not able to obtain sufficient virtual storage to initialize.

**Action:** Increase the amount of above the line region available using the REGION parameter on the EXEC card and restart the region. A REGION=0M value is recommended for the IAMRLS or IAMPLEX address space. The IAMRLS or IAMPLEX region will continue without using the Record Lock Recovery function.

IAML0303 OPEN FAILED FOR RECORD LOCK JOURNAL RC=XX ACBERFLG=XX

**Reason:** The open of the Record Lock Journal file failed with the indicated error codes. The codes are documented in Section 80.22 of the IAM Manual. The IAM Record Lock Recovery function is disabled in the IAMRLS or IAMPLEX address space that encountered this error.

**Action:** To resolve the problem, stop the affected IAMRLS or IAMPLEX address space, and take appropriate corrective action based on the error that was encountered. (See Section 80.22 of the IAM Manual for suggested actions.) Once corrected, restart the failed IAMRLS or IAMPLEX region. Contact Innovation for assistance if needed.

IAML0304 I/O FAILED FOR RECORD LOCK JOURNAL RC=XX RPLERRCD=XX

**Reason:** An I/O request failed on the Record Lock journal dataset, with the indicated return code and error code. These codes are documented in Section 80.21 of the IAM Manual. The Record Lock Recovery function is disabled. The IAMRLS or IAMPLEX address space will continue to run.

**Action:** To reestablish Record Lock Recovery, correct the cause of the error and recycle the IAMRLS or IAMPLEX address space to restart the Record Lock Recovery function.
IAML0305  UNABLE TO ESTABLISH CHECKPOINT EXIT, STIMERM FAILED WITH RC=XX

**Reason:** The request to establish an timed exit to take the periodic check point for the Record Lock Recovery function failed with the indicated return code. Codes are documented in the IBM z/OS assembler macro manual. The Record Lock Recovery process is disabled.

**Action:** Contact Innovation for support. The IAMRLS or IAMPLEX address space will need to be recycled to restart Record Lock Recovery processing.

IAML0306  RECORD LOCK RECOVERY CHECKPOINT COMPLETE

**LOGGER BLOCK:** XXXXXXXX XXXXXXXX or **JOURNAL TTTR:** XXXXXXXX

**Reason:** Informational message that appears periodically when MESSAGES=ALL has been specified that indicates a Record Lock Recovery checkpoint has occurred.

**Action:** No action required, this is an informational only message.

IAML0307  FORMATTING RECORD LOCK RECOVERY DATASET

**Reason:** During IAMRLS or IAMPLEX region startup, the Record Lock Recovery journal dataset is being pre-formatted. This occurs either when it is found to be unloaded or when the LOCKSTART=COLD parameter was specified. Upon completion, the Record Lock Recovery function will be activated.

**Action:** No action required, this is an informational message.

IAML0308  FORMATTED nnn,nnn,nnn OWNER RECORDS AND nnn,nnn,nnn LOCK RECORDS

**Reason:** During IAMRLS or IAMPLEX region startup, it was found that either the record lock recovery dataset was not pre-formatted, or LOCKSTART=COLD was specified. This has resulted in Record Lock Recovery formatting out the journal file it will use to track the lock ownership on recoverable datasets.

**Action:** None, informational only.

IAML0310  ACTIVE LOCK OWNER RECORDS PROCESSED = nnn,nnn,nnn

**Reason:** During IAMRLS or IAMPLEX region startup, the indicated number of lock owners were reestablished.

**Action:** None, informational message only.

IAML0311  ACTIVE RECORD LOCK RECORDS PROCESSED= nnn,nnn,nnn

**Reason:** During IAMRLS or IAMPLEX region startup, the indicated number of record lock were processed and reestablished from the Record Lock journal information.

**Action:** None, informational message only.
IAML0312  LOCK RECORDS DROPPED, MISSING OWNER = nnn,nnn,nnn

**Reason:** During IAMRLS or IAMPLEX region startup, the indicated number of record locks could not be reestablished due to the absence of the corresponding lock owner information, i.e. job name, step name, job id, etc... The most likely reason is that some of the data blocks may not have been physically written out to the record lock journal.

**Action:** If the record lock recovery also reads the IAMRLS / IAMPLEX journal then this message is informational only. However if no journal recovery is performed and the value is not zero, notify contact Innovation support.

IAML0313  RECORD LOCKS RECOVERED AND RETAINED = nnn,nnn,nnn

**Reason:** Indicates the number of record locks that have been recovered from the record lock recovery dataset.

**Action:** Use the display retained locks command to find out what jobs and CICS regions need recovery to be performed to release the recovered locks.

IAML0314  RECORD LOCKS RECORDED AT SHUTDOWN, NO JOURNAL RECOVERY

**Reason:** The record lock recovery function determined that the lock recovery dataset was properly closed when the IAMRLS or IAMPLEX region last terminated. Therefore it will have the proper status of the record locks, and lock recovery does not need to read the IAMRLS or IAMPLEX journal files to update the status of the recovered locks.

**Action:** None, informational message only.

IAML0315  RECORD LOCK RECOVERY INDICATES FAILED STATUS, WILL RE-INITIALIZE DATASET

**Reason:** The record lock recovery encountered some problem resulting in the shut down of the record lock recovery function. For that reason the current data in the lock recovery dataset is not current, so the dataset will be initialized.

**Action:** No action required. If you contact Innovation for support, please go through the RLSLOGDD and job log for any messages that will indicate the original failure that caused record lock recovery to terminate.

IAML0316  IXGBRWSE nnnnn FAILED, RC=XX REAS=XXXXXXXX

**Reason:** The record lock recovery function encountered a failure while processing the system logger journal dataset. The nnnnn is replaced with the function attempted. The request failed with the indicated return code and reason code.

**Action:** Record lock recovery will stop processing the system logger, and use whatever locks have already been recovered. One can use the display retained locks to determine what locks have been recovered and are actively holding records. Contact Innovation for technical support.
IAML0317 UNABLE TO OBTAIN STORAGE TO READ JOURNAL RECORDS

**Reason:** Record lock recovery was unable to obtain sufficient storage to read records from the journal dataset.

**Action:** Record lock recovery will continue without performing the recovery of record locks based on the data written to the record lock dataset only. If this occurs, please take a dump and contact Innovation and FTP the dump to Innovation’s site.

IAML0318 LOCK RECOVERY JOURNAL RECORDS READ = nnn,nnn,nnn

**Reason:** Indicates the total number of journal records read from the time of the last syncpoint until the failure to restore the record locks as they were at the time of the failure.

**Action:** None, informational message only.

IAML0319 LOCK RECOVERY JOURNAL RECORDS USED = nnn,nnn,nnn

**Reason:** Indicates the number of journal records that were actually used to update the record locks.

**Action:** None, informational message only.

IAML0320 LOCK RECOVERY JOURNAL ADDED LOCKS = nnn,nnn,nnn

**Reason:** Indicates the number of additional record locks acquired based on the before image records from the journal.

**Action:** None, informational message only.

IAML0321 LOCK RECOVERY JOURNAL RELEASED LOCKS = nnn,nnn,nnn

**Reason:** Indicates the number of record locks released based on the syncpoint and job step end records read from the journal.

**Action:** None, informational message only.

IAML0322 LOCK RECOVERY JOURNAL ADDED OWNERS = nnn,nnn,nnn

**Reason:** Indicates the number of new lock owners acquired based on the data read from the journal.

**Action:** None, informational message only.

IAML0323 LOCK RECOVERY JOURNAL RELEASED OWNER = nnn,nnn,nnn

**Reason:** Indicates the number of lock owners removed based on the job step end or transaction end data read from the journal.

**Action:** None, informational message only.
IAML0324  LOCK RECOVERY JOURNAL INVALID RECORD= nnn,nnn,nnn

Reason: Indicates the number of journal records that were not recognized as being written by IAM RLS or IAM PLEX address space in the journal.

Action: Use the IAMJUTIL utility (Section 48) SCAN and / or DUMP command to verify the journal dataset or system logger being used for valid records. Then contact Innovation for support.

IAML0325  UNABLE TO OPEN A JOURNAL DATASET FOR LOCK RECOVERY

Reason: An IAM RLS or IAM PLEX address space on a warm start was attempting to read the journal dataset, but for some reason the open failed.

Action: Check the job log for other messages indicating the reason that the journal dataset could not be opened. Record lock recovery will continue with what it has found so far for record locks that it found in the record lock control dataset. If you can correct the problem, you will need to shut down the IAM RLS or IAM PLEX region, restore the lock recovery dataset as of the prior failure, then restart it to redo the processing after correcting the problem with the journal.

IAML0326  LOCK RECOVERY PROCESSING JOURNAL DSN= dataset name

Reason: This message indicates which journal dataset is being used to find any update to record locks.

Action: None, informational message only.

IAML0327  LOCK RECOVERY DJB SCHEDULED J=jobname S=stepname I=jobid

Reason: On the restart of an IAM RLS or IAM PLEX region, the record lock recovery function triggered dynamic job back out recovery for a job step that was processing at the time that the IAM RLS or IAM PLEX region became unavailable.

Action: None, informational message only.

IAML0328  TOTAL RECORD LOCKS RETAINED = nnn,nnn,nnn

Reason: Indicates the total number of record locks reestablished as a result of the record lock recovery function.

Action: None, informational message only.

IAML0386  LOCK RECOVERY SYNAD: I/O error description

Reason: The indicated I/O error occurred while the record lock recovery process was attempting to read the journal file to reestablish record locks.

Action: Record lock recovery will stop reading the indicated journal dataset, and continue to process a subsequent journal dataset if available. This error may result in an incomplete view of the record locks held at the time that the IAM RLS or IAM PLEX region became unavailable. If the error can be corrected, then restart the IAM RLS or IAM PLEX region after restoring the record lock recovery dataset.
**IAML0387  LOCK RECOVERY ALLOC FOR JOURNAL FAILED, RC= nnnn  REAS= xxxx  INFO= xxxx**

**Reason:** The dynamic allocation of a journal file for record lock recovery purposes failed with the indicated error codes.

**Action:** Record lock recovery not read any additional journal datasets. It will have the locks that it had previously, which may not reflect all of the record locks. If the error can be corrected, then shut down the IAM RLS or IAM PLEX region, correct the error, restore the record lock recovery control dataset then restart it. Contact Innovation if additional support is needed.

**IAML0388  UNABLE TO RECOVER LOCK DATA FROM JOURNALS, REQUIRED DATA IS NOT AVAILABLE**

**Reason:** No journal datasets could be found for the IAM RLS or IAM PLEX address space.

**Action:** One or more of the journal datasets could not be found. Record lock recovery will not read any additional journal datasets. It will have the locks that it had previously, which may not reflect all the record lock status when the region became unavailable. If the error can be corrected, then shut down the IAM RLS or IAM PLEX region, correct the error, restore the record lock recovery control dataset then restart it. Contact Innovation if additional support is needed.

**IAML0389  RECORD LOCK RECOVERY ERROR, EXPECTED LOCK SYNONYM POINTER IS 0**

**Reason:** An unexpected problem occurred in the record lock structure as it was being recreated by the record lock recovery process while processing the journal records.

**Action:** Record lock recovery will deactivate itself. The record locks it has reestablished may not reflect the actual record lock circumstance at the time that the region was originally not available. Take a dump of the region using the F iamrls,DUMP command. If you need the record lock recovery, backup the record lock dataset then recycle the IAM RLS or IAM PLEX region with LOKSTART=COLD. Contact Innovation for support.

**IAML0390  JOURNALING NOT ACTIVE, REQUIRED FOR RECORD LOCKING TO FUNCTION**

**Reason:** Record lock recovery did not find any active journaling, to either dasd files or to the system logger. Journaling is necessary for the functioning of record lock recovery to determine the status of record locks since the last record lock recovery syncpoint. Record lock recovery has already set up the status for records locks it found in the record lock dataset, but that may not be the true lock status when the region became unavailable.

**Action:** The record lock recovery function will not be active. If the above error can be resolved, then the record lock recovery dataset should be restored to it’s contents prior to this startup, and it should be recycled.
**IAML0391 RECORD LOCK RECOVERY CHECKPOINT ABENDED**

**Reason:** The record lock recovery code has abended. There should be preceding messages indicating the abend code. Depending on the circumstance, and what recovery can be done, it may have to disable record lock recovery until the region is restarted.

**Action:** Please retain any dumps that have been taken and contact Innovation for support.

---

**IAML0392 UNEXPECTED RECORD LOCK CONTENTION DURING LOCK RECOVERY**

**Reason:** Record lock recovery should be able to obtain all of the record locks were held when the region was still active. Therefore not being able to do so indicates a some type of problem with the record lock recovery function.

**Action:** The record lock recovery function will be disabled until the region is restarted. Please take a dump of the IAM RLS or IAM PLEX region, and make a back up of the record lock control dataset. Then recycle the region with LOCKSTART=COLD specified. Contact Innovation for support.

---

**IAML0393 INSUFFICIENT STORAGE FOR LOCKS DURING LOCK RECOVERY**

**Reason:** IAM was not able to obtain sufficient storage for all of the record locks being recovered.

**Action:** Increase the amount of storage available for record locking. If using 64-bit storage increase the MEMLIMIT default or on the EXEC card. If using 31-bit virtual storage either set REGION=0M, or of that is already the case, use 64-bit virtual storage for the record locks. If those actions do not resolve the problem, then take a dump of the IAM RLS or IAM PLEX region and contact Innovation for support.

---

**IAML0394 RECORD LOCK RECOVERY INVALID LOCK OWNER**

**Lock DSN= dataset name**

**Reason:** The lock record had a reference to a non-existent or invalid owner record.

**Action:** The lock is not re-established. Record lock recovery startup processing will continue. When the region is initialized, take a dump of the region and make a backup copy of the record lock control dataset, then contact Innovation for support.

---

**IAML0395 RECORD LOCK RECOVERY EXCEEDED 10 MISSING OR INVALID OWNER DATA IN LOCK RECORDS**

**Reason:** Too many invalid lock records have been encountered which indicates that there is a potential problem with the record lock recovery dataset.

**Action:** Record Lock Recovery function will be disabled. Take a dump of the IAM RLS or IAM PLEX region that encountered this error, and make a backup copy of the record lock recovery dataset. Then restart the region with LOCKSTART=COLD specified. Contact Innovation for support.
**IAML0396 RECORD LOCK RECOVERY OID INDEX OVERFLOW**

**Reason:** An unexpected error in that several invalid record lock owner OID records were read from the Record Lock Recovery dataset.

**Action:** Record Lock Recovery function will be disabled. Take a dump of the IAM RLS or IAM PLEX region that encountered this error, and make a backup copy of the record lock recovery dataset. Then restart the region with LOCKSTART=COLD specified. Contact Innovation for support.

**IAML0397 RECORD LOCK RECOVERY PROCESSING FAILED**

**Reason:** Record Lock Recovery function encountered a problem which should be described by prior error message(s), and will be now flagged as not active.

**Action:** Record Lock Recovery function will be disabled. Take a dump of the IAM RLS or IAM PLEX region that encountered this error, and make a backup copy of the record lock recovery dataset. If possible, correct the cause of the problem then recycle the IAM RLS or IAM PLEX region. It might be necessary to restart the region with LOCKSTART=COLD. Contact Innovation for support.

**IAML0398 RECORD LOCK RECOVERY INITIALIZATION FAILED**

**Reason:** Record Lock Recovery failed prior to reaching full activation. Prior messages should indicate the reason for the error.

**Action:** Record Lock Recovery function will be disabled. Take a dump of the IAM RLS or IAM PLEX region that encountered this error, and make a backup copy of the record lock recovery dataset. Correct the cause of the problem, then restart the region. It may be necessary to restart with LOCKSTART=COLD specified. Contact Innovation for support.

**IAML0399 RECORD LOCK RECOVERY NOT ACTIVE**

**Reason:** The Record Lock Recovery function has encountered some error that has resulted in the function being disabled.

**Action:** Correct the problem which should be identified in prior messages, then recycle the IAM RLS or IAM PLEX region to reactivate Record Lock Recovery. If necessary contact Innovation for support.

**IAML0502 GETMAIN1 FAILED, JOURNAL PROCESSING TERMINATED**

**Reason:** A getmain for the journal work area failed during journal processing initialization, journal processing is terminated.

**Action:** If journal processing is desired, make sure the IAMRLS address space has a big enough region size to satisfy its storage needs and restart the address space.

**IAML0504 BUFFER GETMAIN FAILED, JOURNAL PROCESSING TERMINATED**

**Reason:** A getmain for the journal output buffers failed during journal processing initialization, journal processing is terminated.
**Action:** If journal processing is desired, make sure the IAMRLS address space has a big enough region size to satisfy it's storage needs and restart the address space.

**IAML0506 SYNAD MESSAGE**

**Reason:** The synad exit was entered during I/O to a journal file.

**Action:** Call Innovation for help in diagnosing this problem.

**IAML0507 LOCATE FAILED FOR JOURNAL DSN dsname**

**Reason:** A locate failed for the specified journal dataset during the initialization process for journaling.

**Action:** Make sure the dataset names provided in the startup parameters are correct and that the datasets are properly cataloged.

**IAML0508 OBTAIN FAILED FOR JOURNAL DSN dsname**

**Reason:** An obtain failed for the specified journal dataset during the initialization process for journaling.

**Action:** Make sure the dataset names provided in the startup parameters are correct and that the dataset(s) exist on the volumes to which they are cataloged.

**IAML0510 IAMBJRNL CLOSED DSN - DDD.DDD.DDD**

**IAML0511 IAMBJRNL USING DSN - dsname**

**Reason:** The journal processor has selected the specified dataset for journal use.

**Action:** None, informational message only.

**IAML0512 JOURNALING HAS BEEN DISABLED**

**Reason:** Journaling has been disabled for IAMRLS processing either due to an initialization error or because no journal datasets have been provided and SYSLOGGER has not been selected.

**Action:** If journaling is desired, fix the initialization error or specify journal datasets or SYSLOGGER in the IAMRLS startup parameters.

**IAML0513 CONNECT FAILED TO LOGSTREAM: logstreamid**

**Reason:** IAMBJRNL was unable to connect to the specified journal logstream.

**Action:** Make sure the system coupling facility policies have been defined correctly and the SYSTEM LOGGER is available on your CPU.

**IAML0514 INIT OF JOURNAL FILES FAILED**

**Reason:** IAMBJRNL was unable to complete initialization of the journal files.
**Action:** Make sure the dataset names provided in the IAMRLS parmlib member for journal datasets is correct, and that they exist on dasd and have available space.

**IAML0515 IAMBJRNL THE NEXT JOURNAL FILE IS NOT EMPTY, JOURNAL FILE SWITCH HAS FAILED.**

**Reason:** A JSWITCH command was entered to force a journal switch or a journal dataset has filled causing an automatic switch, but the next journal dataset is not empty.

**Action:** This will be followed by message IAML0534 requesting you ready the next journal dataset and respond "U" or reply "Q" to terminate the IAMRLS address space.

**IAML0516 IAMBJRNL DSN - dsname DEALLOCATED**

**Reason:** A journal dataset was deallocated.

**Action:** The data in the indicated journal should be copied to another dataset on tape or dasd, and the indicated journal dataset should then be emptied.

**IAML0517 IAMBJRNL DEALLOCATION FAILED FOR DSN- dsname**

**Reason:** The deallocation of a journal dataset failed.

**Action:** Call Innovation for help in diagnosing this problem.

**IAML0518 IAMBJRNL DSN - dsname ALLOCATED**

**Reason:** A Journal dataset has been allocated for use.

**Action:** None, informational message only.

**IAML0519 IAMBJRNL ALLOCATION FAILED FOR DSN- dsname**

**Reason:** The allocation of a journal dataset has failed.

**Action:** IAMRLS should switch to the next journal dataset if possible. Call Innovation for help in diagnosing this problem.

**IAML0520 IAMBJRNL CONNECTED TO LOGSTREAM: logstream**

**Reason:** The name of the System Logger Logstream that IAMRLS connected to for journal processing is displayed.

**Action:** None, informational message only.

**IAML0521 IAMBJRNL DYNALLOC ERROR, RC= , S99ERROR= , S99INFO=**

**Reason:** IAMBJRNL received an error during dynamic allocation or deallocation or a journal dataset.

**Action:** None.
IAML0524 IAMDD ddname: SPANNED RECORD TOO BIG, JOURNAL PROCESSING TERMINATED.

Reason: Journaling was attempted on a spanned record file, but the record was larger then 32,756 which is the maximum size allowed.

Action: Turn off journaling for this file, it is not allowed with such a large record size.

IAML0525 IAMDD ddname: SHUTDOWN IN PROGRESS OR JOURNAL PROCESSING TERMINATED

Reason: The journal exit tried to journal a request for IAMDD ddname and was unsuccessful due to IAMRLS shutting down or the IAMRLS journal processing was terminated.

Action: Check the RLSLOGDD and JESLOG for other messages indicating why journal processing has been terminated or why IAMRLS is shutting down. Call Innovation for help in diagnosing this problem.

IAML0526 IAMDD ddname: GETMAIN1 FAILED, JOURNAL PROCESSING TERMINATED

Reason: The journal exit, IAMJRNRL, could not obtain storage for a journal work area for IAMDD ddname.

Action: Journaling for this dd is terminated. Make sure that the IAMRLS address space has enough region size specified at startup time.

IAML0527 IAMDD ddname: SAVEAREA GETMAIN FAILED, JOURNAL PROCESSING TERMINATED

Reason: The journal exit, IAMJRNRL, could not obtain storage for a save area to schedule a journal request to the journal subtask.

Action: Journaling for this dd is terminated. Make sure that the IAMRLS address space has enough region size specified at startup time.

IAML0528 IAMDD ddname: IXGWRITE FAILED, RC=nn, RSNCD=nnnnn

Reason: The journal exit has received a failing return code from the System Logger during a write operation. This message will only appear once for every 30 errors to aid so the console will not be flooded with messages.

Action: RSNCD=00000860 indicates that an offload is needed, which should be automatically handled by the system. If it is not, one can use the IAM/PLEX command to offload the logger (F liamplex1,LOGGER,OFFLOAD). When the offload has completed message IAM0612 will come out and the logger processing will continue. For other codes call Innovation for help in diagnosing this problem. You can use the IAMPLEX LOGGER commands to disconnect from the current logger, then define a new one, update the IAM/PLEX parameters, and start connect the new logger dataset.

IAML0529 IAMDD ddname: JOURNAL UPDATE FAILED

Reason: The journal subtasks work area could not be found to schedule a journal request for IAMDD ddname.
Action: The journal request is not scheduled and the journal entry is lost. Contact Innovation for help in diagnosing this error.

IAML0530 IAMDD ddname: JOURNAL PROCESSING TERMINATED, JOURNAL ENTRIES LOST.

Reason: There is a problem with the journal buffer queue and a journal entry has not been able to be written.

Action: Check the RLSLOGDD and JESLOG for other messages indicating the source of the journaling problem. Call Innovation for help in diagnosing this error.

IAML0531 IAMDD ddname: IXGWRITE FAILING, JOURNAL PROCESSING TERMINATED

Reason: The journal exit is receiving failing return codes from the System Logger. Journal processing is terminated.

Action: Call Innovation for help in diagnosing this problem.

IAML0533 IAMBJRNL INITIALIZATION OF JOURNAL FILES FAILED, IAMRLS WILL QUIESCE.

Reason: The journal subtask was unable to initialize any of the journal datasets provided in the IAMRLS parmlib.

Action: Make sure the dataset names provided in the IAMRLS parmlib are correct, that the datasets exist on dasd and have available space.

IAML0534 READY DSN dsname, THEN REPLY 'U', OR REPLY 'Q' TO TERMINATE IAMRLS

Reason: The next journal dataset was not empty during a switch operation.

Action: Empty dataset dsname, then reply 'U' to continue journal processing, or reply 'Q' to terminate the IAMRLS address space.

IAML0535 ONLY 1 JOURNAL DATASET WAS PROVIDED, REPLY 'U' TO REUSE THE SAME DATASET, OR REPLY 'Q' TO TERMINATE IAMRLS.

Reason: The single journal dataset has been filled. IAMRLS needs operator response to start reusing the dataset in order to preserve the data. All I/O requests that need to be journaled will be delayed until a response is provided.

Action: Copy off the data from the journal dataset, then reply U to continue processing. Consider adding additional journal datasets to the IAMRLS region so that delays such as this can be eliminated.

IAML0536 OFFLOAD PROCESSING FOR LOGSTREAM nnnnnnnn.nnnnnn COMPLETED, RC=xxxx,RSN=xxxx

Reason: The requested offload process for the logstream journal has completed with the indicated return and reason codes.

Action: None if the return code is 0. If the return code is other than 0, determine the cause of the problem. Contact Innovation technical support if assistance is needed.
IAML0551 IAMSYNC STORAGE OBTAIN FAILED, NO SYNCPOINT RECORD WRITTEN

Reason: IAMSYNC was unable to obtain storage for a work area and will not be able to write the syncpoint record for the batch program that called it.

Action: Make sure the batch job has sufficient region size to satisfy the storage request.

IAML0552 IAMSYNC ATTEMPT TO LOAD IAMASPT FAILED, NO SYNCPOINT RECORD WRITTEN

Reason: IAMSYNC was unable to load module IAMASPT to perform syncpoint processing.

Action: Check the LINKLIST/JOBLIB/STEPLIB and make sure that the batch job has access to the IAM load library where module IAMASPT is located.

IAML0561 IAMASPT UNABLE TO LOAD IEANTRT, NO SYNCPOINT RECORD WRITTEN

Reason: IAMASPT was unable to load the IBM module IEANTRT to perform name token services. Without this module IAMASPT cannot write a syncpoint record.

Action: Make sure that IBM's common services library, SYS1.CSSLIB, is in the LINKLIST or that programs have access to it.

IAML0562 IAMASPT RECEIVED RETURN CODE nnn FROM TOKEN SERVICES, NO SYNCPOINT RECORD WRITTEN

Reason: IAMASPT was unable to retrieve the Global Token Area for the IAMRLS address space and cannot perform SYNCPOINT services.

Action: Check to see that IAMRLS is active, if not start it and restart the job. If it is active, call Innovation for help in diagnosing this problem.

IAML0563 IAMASPT RECEIVED ETURN CODE nnn FROM ALESERV EXTRACT, NO SYNCPOINT RECORD WRITTEN

Reason: The IAM batch syncpoint process encountered an error when invoking the IBM ALESERV service.

Action: Contact Innovation for assistance.

IAML0564 IAMASPT HAS TRIED TO EXCEED THE LIMIT OF 103 FILES, NO SYNCPOINT RECORD WRITTEN

Reason: The IAM batch syncpoint process has a limit of 103 IAM files.

Action: Either reduce the number of files that the program is using, or contact Innovation for assistance.
IAM0565 IAMASPT FOUND IAM RLS MARKED AS INACTIVE IN THE GLOBAL TOKEN Area, NO SYNCPOINT RECORD WRITTEN

Reason: IAMRLS has been marked as inactive and no syncpoint services are available.

Action: Determine why the IAMRLS address space is inactive and if possible, restart IAM RLS then restart the batch job.

IAM0566 IAMASPT FOUND IAM RLS MARKED AS QUIESCE FORCE IN THE GLOBAL TOKEN, NO SYNCPOINT RECORD WRITTEN

Reason: A QUIESCE,FORCE condition exists in the IAMRLS address space and it is longer accepting requests.

Action: Determine why the IAMRLS address space has been quiesced.

IAM0567 IAMASPT RECEIVED ETUM CODE nnn FROM THE RLS JOURNAL CALL, NO SYNCPOINT RECORD WRITTEN

Reason: The syncpoint writer in RLS encountered an error.

Action: Check this job log and the RLS job log for other error messages and contact Innovation for assistance.

IAM0571 IAMBSYNC UNABLE TO OBTAIN STORAGE; NO SYNCPOINT WRITTEN

Reason: IAMBSYNC was unable to obtain storage for a work area, the syncpoint record could not be written.

Action: Make sure the job has a large enough region size when started to accommodate all of the storage required.

IAM0572 IAMBSYNC FOUND JOURNAL-IN-TRouble FLAG ON IN THE IWA, NO SYNCPOINT WRITTEN

Reason: The journal subtask has set a bit in the IWA indicating that initialization failed or that journaling is not active. The syncpoint record cannot be written.

Action: Browse the RLSLOGDD and JESLOG for additional messages to determine why the journal subtask is not active.

IAM0573 IAMBSYNC FOUND AN ADDRESS OF ZERO IN THE GLOBAL TOKEN FOR THE JOURNAL’S TWA, NO SYNCPOINT WRITTEN

Reason: There was no task work area for the journal subtask in the Global Token Area. The subtask has either failed or was not activated.

Action: Browse the RLSLOGDD and JESLOG for additional messages to determine why the journal subtask has failed or is not active.
IAML0574 IAMBSYNC FOUND JOURNAL FULL BIT ON AFTER WAITING FOR THE JOURNAL WRITE, NO SYNCPOINT WRITTEN

Reason: There is a problem with the journal output buffers, the syncpoint record cannot be written.

Action: Call Innovation for help in diagnosing this problem.

IAML0575 IAMBSYNC IXWRITE FAILED; RC=nn, RSNCD=nnnn, NO SYNCPOINT RECORD WRITTEN

Reason: IAMBSYNC attempted to write a syncpoint record to the System Logger, and the write failed with the indicated return and reason codes.

Action: Browse the RLSLOGDD and JESLOG for additional messages that might indicate why the System Logger Connection is not working. Call Innovation for help in diagnosing this problem.

IAML0576 IAMBSYNC’S CALL TO RELEASE THE LOCKS HELD BY THIS JOB FAILED

Reason: A call to release the record locks held by this job has failed.

Action: Call Innovation for help in diagnosing this problem.

IAML0577 IAMBSYNC FOUND AN INVALID OCQ IN THE PARAMETER LIST, OCQ BYPASSED

Reason: The batch syncpoint process encountered an invalid pointer to an expected open dataset. Processing will continue with the next open dataset for this job, if any.

Action: Contact Innovation technical support.

IAML0578 IAMBSYNC IXCMMSGO MESSAGE NOT ACCEPTED SYNCPOINT NOT COMPLETED

Reason: The attempt by a router IAM/PLEX to send an request to a target IAM/PLEX for syncpoint processing has failed.

Action: Contact Innovation technical support.

IAML0579 IAMBSYNC IXCMMSGO RESPONSE MESSAGE NOT ACCEPTED SYNCPOINT NOT COMPLETED

Reason: The target IAM/PLEX address space failed notifying the router IAM/PLEX that it had completed syncpoint processing.

Action: Contact Innovation technical support.

IAML0580 IAMBSYNC’S CALL TO FIND THE OID FOR THIS JOB FAILED

Reason: Syncpoint processing attempted to find the control information for the job that issued the syncpoint, but was unable to do so.

Action: Contact Innovation technical support.
IAML0581 IAMBIJNL UNABLE TO OBTAIN STORAGE, NO BATCH JOURNAL PROCESSING WILL TAKE PLACE

Reason: The batch journal task was unable to obtain storage and no batch journal processing can happen.

Action: Restart IAM RLS with more storage.

IAML0582 IAMBIJNL FOUND JOURNAL-IN-TROUBLE FLAG ON IN THE TWA; NO JOURNAL RECORD WRITTEN

Reason: The batch journal task was found an unexpected flag value and cannot proceed.

Action: Dump the IAM RLS address space and contact Innovation.

IAML0583 IAMBIJNL FOUND A ZERO ADDRESS IN THE GLOBAL TOKEN FOR THE JOURNAL'S TWA, NO JOURNAL WRITTEN

Reason: The batch journal task cannot process without a valid address for the journal task's work area.

Action: Dump the IAM RLS address space and contact Innovation.

IAML0584 IAMBIJNL FOUND JOURNAL FULL BIT ON AFTER WAITING FOR JOURNAL WRITE, NO JOURNAL RECORD WRITTEN

Reason: The batch journal task has encountered a journal full condition after waiting for the journal task.

Action: Dump the IAM RLS address space and contact Innovation.

IAML0585 IAMBIJNL IXGWRITE FAILED; RC=.., RSNCD=........, NO JOURNAL RECORD WRITTEN

Reason: The batch journal task has received a non-zero return code from the system log write.

Action: Dump the IAM RLS address space and contact Innovation.

IAML0600 IAMBSYNC IXGWRITE FAILED, RC= , RSNcd= , WILL WAIT FOR RETRY

Reason: IAMBSYNC attempted to write a syncpoint record to the SYSTEM LOGGER and the write failed with the specified return and reason codes. The write will be retried after the SYSTEM LOGGER indicates the condition is resolved.

Action: None. Should see the IAML0601 message after retry.

IAML0601 IAMBSYNC IXGWRITE REISSUED AFTER LOGGER RECOVERY.

Reason: IAMBSYNC was notified of a SYSTEM LOGGER recovery and the failed write of the syncpoint record was reissued.

Action: None.
IAML0602 IAMBSYNC CALLED FOR CICS SYNCPOINT
Reason: IAMBSYNC was called for a CICS syncpoint.
Action: None.

IAML0603 IAMBSYNC CALLED FOR EXTERNAL CICS SYNCPOINT
Reason: IAMBSYNC was called for an external CICS syncpoint.
Action: None.

IAML0604 IAMBJRNL DISCONNECTED FROM LOGSTREAM: logstream
Reason: IAMBJRNL has disconnected from the logstream named.
Action: None.

IAML0605 IXGWRITE OF INIT RECORD FAILED, RC= ,RSNCD=
Reason: IAMBJRNL attempted to write an initialization record to the logstream and the write failed with the specified codes.
Action: Contact Innovation for support.

IAML0606 LOGGER COMMAND COULD NOT BE PROCESSED
Reason: The LOGGER command was issued but the System Logger is not being used in this RLS or PLEX region. The command is ignored.
Action: None.

IAML0607 LOGGER COMMAND MISSING COMMAND PARM
Reason: The LOGGER command was issued without a DISCON, RECON, POST, or OFFL parameter. The command is ignored.
Action: Reissue the command with the appropriate parameter.

IAML0608 OPEN FILES HAVE BEEN DETECTED, REPLY 'Y' TO FORCE SHUTDOWN, OR REPLY 'W' TO WAIT FOR FILES TO CLOSE
Reason: A QUIESCE command was issued but there are still some open files.
Action: Reply with 'Y' to force shutdown of the region or reply 'W' to wait for the files to close.

IAML0609 IAMZAPR COULD NOT BE LOADED
Reason: An APPLY command was issued but module IAMZAPR could not be loaded.
Action: Contact Innovation for support.
IAML0610 IAMZAPR COULD NOT BE LOADED

**Reason:** A RESTORE command was issued but module IAMZAPR could not be loaded.

**Action:** Contact Innovation for support.

IAML0611 NO OTHER MEMBERS HAVE CONNECTED TO THIS GROUP

**Reason:** A Display Group command was issued, and no other members are connected to this RLSGROUP.

**Action:** None.

IAML0612 BJRNLSRBP POSTED BJRNLS FOR RECOVERY AFTER WRITE ERROR

**Reason:** An offload or other recovery operation has been completed by the SYSTEM LOGGER on the logstream used by this RLS and the BJRNLSRBP task has been posted to retry any writes that may have failed during the recovery operation.

**Action:** None.

IAML1000 IAMRLS/CICS Initialization complete. IAM RLS Support active vn.n/mlP (xxxxxxxx)

**Reason:** Informational message indicating that IAM CICS exits have successfully completed initialization and shows the Version release modification level of the IAM code.

**Action:** None

IAML1001 IAMRLS/CICS is connected to IAM RLS with RLSID = <id>
IAMRLS/CICS has disconnected from IAM RLS with RLSID = <id>
IAMRLS/CICS is reconnected to IAM RLS with RLSID = <id>

**Reason:** Informational message showing that the IAM CICS exits have successfully connected, disconnected, or reconnected to the IAM/RLS (or IAM/PLEX) region which has an RLSID of <id>.

**Action:** None

IAML1002 IAMXCINI - IAM PLT Stage <i> with CICS level: nnnn

where: <stat> = {begun | ended}

<nnnn> = CICS Internal Release level.

**Reason:** Informational message showing that the IAM CICS program IAMXCINI has been invoked by CICS initialization during the specified Phase and what the IAMXCINI program has determined to be the CICS internal Release level.

**Action:** None
IAML1003 IAMXCINI - IAM PLT Stage <i> initialization has <status> or IAM vn.n/mlP.

where:  <i>   = [ II or III ] CICS Initialization phase
<status> = { begun ¦  ended }

Note: Programs in the CICS PLTPI before the DELIM statement get executed during CICS initialization Phase II. While programs after the PLT DELIM statement will get executed during initialization Phase III, also known as final initialization. IAMXCINI requires an entry in the PLT both before and after the PLT DELIM entry.

Reason: Informational message showing that the IAM CICS program IAMXCINI has been invoked by CICS initialization during the specified Phase.

Action: None

IAML1004 IAMXCINI - IAM TRUE exit program IAMBCICS has been ENABLED.

Reason: Informational message showing that the IAM CICS program IAMXCINI has successfully installed the IAM TRUE (task related user exit) program and enabled it for the current CICS execution.

Action: None

IAML1005 IAMXCINI - IAM GLUE exit program IAMXFCBO has been ENABLED at EXIT(XFCBOUT).

Reason: Informational message showing that the IAM CICS program IAMXCINI has successfully installed the IAM GLUE (global user exit) program at the specified CICS exit point and has enabled it for the current CICS execution.

Action: None

IAML1006 IAMXCINI - IAM GLUE exit program IAMXFCBO has been STARTED at EXIT(XFCBOUT).

Reason: Informational message showing that the IAM CICS program IAMXCINI has successfully started the IAM GLUE (global user exit) program at the specified CICS exit point which causes CICS to invoke the exit whenever the specified exit point gets driven.

Action: None

IAML1007 IAMXCINI - IAM TRUE exit program IAMBCICS has been STARTED.

Reason: Informational message showing that the IAM CICS program IAMXCINI has successfully started the IAM TRUE (Task related User exit) program. The IAM exit is fully active when this message occurs.

Action: None
IAM/RSL and IAM/PLEX Messages 80.05

IAML1008 IAMXCINI - IAM GWA-global work area is at: x('xxxxxxxx')

Reason: Informational message showing the address of the IAM Global Work Area (GWA) that gets allocated when the IAM Task related User exit) program is initially "ENABLED". This area is utilized to communicate with an associated IAM/RLS (or IAM/PLEX) address space.

Action: None

IAML1009 IAMXCINI - IAM Initialization ENQUEUE was issued.

Reason: Informational message showing the IAM initialization has issued a z/OS Enqueue within the CICS address space to control and serialize IAM actions within the CICS region.

Action: None

IAML1010 Backout requested for file=<fileid> Task=nnnnn Tran=xxxx UOWID=xxxxxxxx xxxxxxxxx

Reason: Informational message showing the IAM file, task number transaction id and associated UOWID that are about to be backed out due to having incurred an ABEND, or because the executing program issued a SYNCPOINT ROLLBACK or because CICS Emergency Restart determined that there were pending Shunted Units-of-work (UOWs).

Action: None - Informational

IAML1011 IAMXCINI - IAM program: <pgmnam> was loaded at x('xxxxxxxx').

Reason: Informational message showing the address within the CICS region that an IAM program has been loaded by CICS program load services.

Action: None

IAML1012 IAMXCINI - IAM program: <pgmnam> was not loaded.

Reason: Informational message showing that the specified IAM program was not loaded within the CICS address space. This is a normal message as the program may be a feature of IAM not utilized, such as the IAM GLUE IAMXFCAB which is part of the IAMRLS Record Lock Timeout function.

Action: None
IAML1013 IAMXCINI - IAM PLT program is being invoked more than one time during PLT Stage <i>.</i>

where:  <i> = [ II or III ]

**Reason:** Informational message showing that the specified IAM PLT program was invoked by CICS more than once during the specified CICS initialization Phase.

**Action:** Check your PLTPI statements for any duplicate IAM program entities within each section of the PLT (there should be an entry in the PLT for IAMXCINI both before and after the DELIM entry).

IAML1014 IAMXCINI - IAM requires a PLTPI Stage II entry for IAMXCINI with CICS TS 3.1 and above in PLTPI.

**Reason:** Diagnostic message showing that IAM requires both a CICS PLT Phase II and Phase III entry at CICS TS v3.1 and above in order for IAM to fully initialize and function with an associates IAMRLS region.

**Action:** Check your PLTPI statements making sure there are two entities within each section of the PLT (there should be an entry in the PLT for IAMXCINI both before and after the DELIM entry).

IAML1015 IAMXCINI - IAMXFCBO not ENABLED during PLT Stage II; Resp='xxxxxxxx'; Resp2='xxxxxxxx'; EIBRCODE='xxxxxxxx'

**Reason:** Diagnostic message showing that IAM was unable to ENABLE the IAM GLUE exit program IAMXFCBO during PLT Stage II. Without the exit IAMXFCBO enabled, backout and syncpoint calls to an associates IAMRLS region will be fully functional, causing a data integrity exposure.

**Action:** Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. Correct the indicated underlying cause and restart CICS

IAML1016 IAMXCINI - Failed to LOAD the IAM Global Options Table: IAMOPT, IAM is disabled.

**Reason:** Diagnostic message showing that IAM was unable to find the IAM Global Options Table load module "IAMOPT" as is required for the functioning of the IAM CICS Automatic IAMRLS Reconnect feature.

**Action:** Check the CICS Steplib concatenation to make sure that the IAM Global Options Table is within a load module library that is specified current concatenation. The IAM Global Options Table module name is IAMOPT. After correcting the underlying cause and restart CICS.

IAML1017 IAMXCINI - IAM Global Options Table is at: x(nnnnnnnn) Version: (vn.n/mlP).

**Reason:** Informational message showing the address within the CICS region that the IAM Global Options Table has been loaded.

**Action:** None - Informational
IAM/RLS and IAM/PLEX Messages

IAM1020 IAMXCIINI - exit program(IAMBCICS) is already ENABLED

**Reason:** Informational message showing the IAM Task Related Exit program is currently enabled.

**Action:** None - Informational

IAM1021 IAMXCIINI - ILLOGICAL ERROR on INQUIRE PROGRAM <pgmtype>(<pgmname>);
Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'

**Where:** <pgmtype> = { PROGRAM }<pgmname> = one of the IAM exit program names

**Reason:** Diagnostic message showing that IAM was unable to execute the INQUIRE PROGRAM (or EXITPROGRAM) function for the specified program.

**Action:** Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. Correct the indicated underlying cause and restart CICS

IAM1022 IAMXCIINI - NOTAUTH ERROR on INQUIRE PROGRAM( <pgmtype> (<pgmname>);
Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'

**Where:** <pgmtype> = { PROGRAM }<pgmname> = one of the IAM exit program names

**Reason:** Diagnostic message showing that IAM was unable to execute the INQUIRE PROGRAM (or EXITPROGRAM) function for the specified program.

**Action:** Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. Correct the indicated underlying cause and restart CICS

IAM1023 IAMXCIINI - PGMIDERR ERROR on INQUIRE PROGRAM <pgmtype>(<pgmname>);
Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'

**Where:** <pgmtype> = { PROGRAM }<pgmname> = one of the IAM exit program names

**Reason:** Diagnostic message showing that IAM got a PGMIDERR when it did an INQUIRE PROGRAM function for the specified program.

**Action:** Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. Correct the indicated underlying cause and restart CICS
IAML1024  IAMXCINI - UNKNOWN ERROR on INQUIRE <pgmtype>(<pgmname>); Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'

Where: <pgmtype> = { PROGRAM | EXITPROGRAM }
      <pgmname> = one of the IAM exit program names

Reason: Diagnostic message showing that IAM got an UNKNOWN error when it did an INQUIRE PROGRAM (or EXITPROGRAM) function for the specified IAM program.

Action: Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. Correct the indicated underlying cause and then restart the CICS region.

IAML1025  IAMXCINI - IAM exit program: <pgmname> was not ENABLED, invalid options.

Where: <pgmname> = one of the IAM exit program names

Reason: Diagnostic message showing that IAM got an error response when it did an INQUIRE PROGRAM (or EXITPROGRAM) function for the specified IAM program.

Action: Check for an occurrence of an associated IAML1030 message. IAM uses the GWA to communicate with a connected IAMRLS (or IAM/PLEX) region for sharing IAM files. Without this IAM will not be able to fully function. Determine and correct the indicated underlying cause and then restart the CICS region.

IAML1026  IAMXCINI - IAM exit program: <pgmname> has no GWA-global area.

Reason: Diagnostic message showing that IAM got an INVEXTREQ when it did an EXTRACT EXIT PROGRAM function to obtain the address of the IAM GWA associated with the IAM GLUE Exit program.

Action: Check for an occurrence of an associated IAML1030 message. IAM uses the GWA to communicate with a connected IAMRLS (or IAM/PLEX) region for sharing IAM files. Without this IAM will not be able to fully function. Determine and correct the indicated underlying cause and then restart the CICS region.

IAML1027  IAMXCINI - <pgmname> exit load module has changed since exit ENABLED

Reason: Diagnostic message showing that IAM got an INVEXTREQ when it did an EXTRACT EXIT PROGRAM function to obtain the address of the IAM GWA associated with the IAM GLUE Exit program.

Action: Check for an occurrence of an associated IAML1030 message. IAM uses the GWA to communicate with a connected IAMRLS (or IAM/PLEX) region for sharing IAM files. Without this IAM will not be able to fully function. Determine and correct the indicated underlying cause and then restart the CICS region.

IAML1029  IAMXCINI - INVEXITREQ ERROR on ENABLE PROGRAM(<pgmnam>)
Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'

Where: <pgmname> = one of the IAM exit program names
**Reason:** Diagnostic message showing that IAM got an INVEXITREQ when it did an ENABLE PROGRAM function to define an IAM program as a CICS Task Related User Exit.

**Action:** Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. IAM uses the EXIT to communicate with an connected IAMRLS (or IAM/PLEX) region for sharing IAM files. Without this IAM will not be able to fully function. Determine and correct the indicated underlying cause and then restart the CICS region.

**IAML1030 IAMXCINI - INVEXITREQ ERROR on EXTRACT EXIT PROGRAM(<pgmnam>) Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'

**Reason:** Diagnostic message showing that IAM got an INVEXITREQ when it did an EXTRACT EXIT PROGRAM function to obtain the address of the IAM GWA associated with the IAM GLUE Exit program.

**Action:** Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. IAM uses the GWA to communicate with an associated IAMRLS (or IAM/PLEX) region for sharing IAM files. Without this IAM will not be able to fully function. Determine and correct the indicated underlying cause and restart CICS.

**IAML1031 IAMXCINI - INVEXITREQ ERROR on ENABLE PROG(<pgmname>) START Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'

**Where:** <pgmname> = one of the IAM exit program names

**Reason:** Diagnostic message showing that IAM got an INVEXITREQ when it did an ENABLE PROGRAM START function to start the specified IAM exit program.

**Action:** Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. IAM uses the exits to communicate with a connected IAMRLS (or IAM/PLEX) region for sharing IAM files. Without this IAM will not be able to fully function. Determine and correct the indicated underlying cause and then restart the CICS region.

**IAML1032 IAMXCINI - INVEXITREQ ERROR on ENABLE PROG(<pgmname>) EXIT(<exitname>) Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'

**Where:** <exitname> = one of the CICS exit point names
<pgmname> = one of the IAM exit program names

**Reason:** Diagnostic message showing that IAM got an INVEXITREQ when it did an ENABLE PROGRAM EXIT function to enable the specified IAM exit program.

**Action:** Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. IAM uses the exits to communicate with a connected IAMRLS (or IAM/PLEX) region for sharing IAM files. Without this IAM will not be able to fully function. Determine and correct the indicated underlying cause and then restart the CICS region.
IAML1033 IAMXCINI - IAM EXIT PROGRAM(<pgmnam>) is already ENABLED.

Where:  <pgmname>  = one of the IAM exit program names

Reason: Diagnostic message showing that IAM found there to already be an existing program <program> defined to the CICS region as an exit program.

Action: Review the CICS CSD / RDO definitions for a pre-existing definition matching the specified program <pgmname>. All of IAM's CICS programs begin with the prefix IAM and should be treated as a reserved program name. Rename, remove or delete the other non-IAM created definitions corresponding to the specified program <pgmname> and restart CICS

IAML1034 IAMXCINI - IAM EXIT POINT(<pgmnam>) is invalid.

Where:  <exitname> = one of the CICS exit point names

Reason: Diagnostic message showing that IAM got a UNKNOWN error response code when it did an INQUIRE PROGRAM function for the specified program.

Action: Check load libraries for valid load module <prgname> as it may not be an marked as an executable load module. Refresh the load module <pgmname> from the IAM distribution library.

IAML1035 IAMXCINI - IAM EXIT PROGRAM(<pgmnam>) is not defined to CICS.

Where:  <pgmname>  = one of the IAM exit program names

Reason: Diagnostic message showing that IAM got a undefined error response code when it did an INQUIRE PROGRAM function for the specified program.

Action: Check CICS traces or prior messages to determine why the IAM exit program <pgmname> did not get defined.

IAML1036 IAMXCINI - IAMBCICS exit global work area length exceeds maximum.

Reason: Diagnostic message showing that IAM got a length error response code when it attempted to define the IAM Global User Exit (GLUE) program.

Action: Check in the CICS logs and message output for any possible storage violations or overlays. Dump the CICS region and retain the generated dump, contact IAM support at Innovation for further assistance and directions.

IAML1037 IAMXCINI - NOTAUTH ERROR on ENABLE EXIT PROGRAM(<pgmname>)

Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'

Where:  <pgmname>  = one of the IAM exit program names

Reason: Diagnostic message showing that IAM got a NOTAUTH when it did an ENABLE EXIT PROGRAM function to enable an IAM Exit program.

Action: Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. IAM uses the exits to communicate with a connected IAMRLS (or
IAM/PLEX) region for sharing IAM files. Without this IAM will not be able to fully function. Determine and correct the indicated underlying cause and then restart the CICS region.

**IAML1038 IAMXCINI - NOTAUTH ERROR on ENABLE PROG(<pgmname>) START**

*Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'*

**Where:** <pgmname> = one of the IAM exit program names

**Reason:** Diagnostic message showing that IAM got a NOTAUTH when it did an ENABLE EXIT PROGRAM function to enable an IAM Exit program at GWA associated with the IAM GLUE Exit program.

**Action:** Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. IAM uses the exits to communicate with a connected IAMRLS (or IAM/PLEX) region for sharing IAM files. Without this IAM will not be able to fully function. Determine and correct the indicated underlying cause and then restart the CICS region.

**IAML1039 IAMXCINI - NOTAUTH ERROR on EXTRACT EXIT PROGRAM(<progname>)**

*Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'*

**Where:** <pgmname> = one of the IAM exit program names

**Reason:** Diagnostic message showing that IAM got a NOTAUTH when it did an EXTRACT EXIT PROGRAM function to obtain the address of the IAM GWA associated with the IAM GLUE Exit program.

**Action:** Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. IAM uses the GWA to communicate with an associated IAMRLS (or IAM/PLEX) region for sharing IAM files. Without this IAM will not be able to fully function. Determine and correct the indicated underlying cause and then restart the CICS region.

**IAML1040 IAMXCINI - UNKNOWN ERROR on ENABLE EXIT PROG(<pgmname>) EXIT(<exitname>)**

*Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'*

**Where:** <exitname> = one of the CICS exit point names

<pgmname> = one of the IAM exit program names

**Reason:** Diagnostic message showing that IAM got a NOTAUTH when it did an ENABLE EXIT PROGRAM function to enable an IAM Exit program.

**Action:** Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. IAM uses the exits to communicate with a connected IAMRLS (or IAM/PLEX) region for sharing IAM files. Without this IAM will not be able to fully function. Determine and correct the indicated underlying cause and then restart the CICS region.

**IAML1041 IAMXCINI - UNKNOWN ERROR on ENABLE EXIT PROG(<pgmname>) START**

*Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'*

**Where:** <pgmname> = one of the IAM exit program names
**Reason:** Diagnostic message showing that IAM got an UNKNOWN error when it did an ENABLE EXIT PROGRAM function to START an already enabled IAM exit program.

**Action:** Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. IAM uses the exits to communicate with a connected IAMRLS (or IAM/PLEX) region for sharing IAM files. Without this IAM will not be able to fully function. Determine and correct the indicated underlying cause and then restart the CICS region.

IAML1042 IAMXCINI - UNKNOWN ERROR on ENABLE EXIT PROG(<pgmname>)
Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'

Where: <pgmname> = one of the IAM exit program names

**Reason:** Diagnostic message showing that IAM got an UNKNOWN error when it did an ENABLE EXIT PROGRAM function to define an IAM exit program.

**Action:** Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. IAM uses the exits to communicate with a connected IAMRLS (or IAM/PLEX) region for sharing IAM files. Without this IAM will not be able to fully function. Determine and correct the indicated underlying cause and then restart the CICS region.

IAML1043 IAMXCINI - UNKNOWN ERROR on EXTRACT EXIT PROGRAM(<pgmname>)
Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'

where: <pgmname> = one of the IAM exit program names

**Reason:** Diagnostic message showing that IAM got an UNKNOWN error when it did an EXTRACT EXIT PROGRAM function for an IAM exit program.

**Action:** Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. IAM uses the exits to communicate with a connected IAMRLS (or IAM/PLEX) region for sharing IAM files. Without this IAM will not be able to fully function. Determine and correct the indicated underlying cause and then restart the CICS region.

IAML1044 IAMXCINI - IAM abnormal condition occurred. Reply GO or CANCEL

**Reason:** Diagnostic message showing that IAM ran into a situation during initialization and determined it could not continue further with initialization.

**Action:** Reply "GO" to continue with CICS initialization without IAM being fully functional within the CICS region or reply with "CANCEL" to cause a full CICS region dump to be generated. Retain the generated dump, contact IAM support at Innovation for further assistance and directions.

IAML1045 IAMXCINI - An EXEC CICS PERFORM DUMP has been issued.

**Reason:** Diagnostic message showing that IAM ran into a situation that was illogical and determined a dump was necessary.
Action: Review the installation of IAM, reinstalling if necessary within the CICS environment if a discrepancy is uncovered. Retain the generated dump, contact IAM support at Innovation for further assistance and directions.

IAML1050 IAMXCNTL - Processing of <type> request has begun for <request string>

Reason: Informational message showing that the IAM CICS program IAMCNTL has begun to process the specified request.

Action: None

IAML1051 IAMXCNTL - Processing of <type> request has completed for <request string>

Reason: Informational message showing that the IAM CICS program IAMCNTL has completed processing the specified request.

Action: None

IAML1052 IAMXCNTL - Retrying Shunted UOW for DSN(<dataset name>)

Reason: Informational message showing that the IAM CICS program IAMCNTL is attempting to RETRY shunted UOW(s) for the specified dataset.

Action: None

IAML1053 IAMXCNTL - Can not re-open FILE(<fileid>) for DSN(<dataset name>)

Reason: Informational message showing that the IAM CICS program IAMXCNTL has determined that it can not re-open the FILE.

Action: Check the CICS job and messages logs for further error messages. The affected dataset may required direct manual operator intervention via the CEMT transaction to RETRY any Shunted UOWs against the specified DATASET and then to issue an OPEN request for the specified FILEID.

IAML1054 IAMXCNTL - DSN(<dataset name>) still has pending SHUNTED UOW(s)

Reason: Informational message showing that the IAM CICS program IAMXCNTL has determined that there is at least one UOWENQ pending for the specified dataset. IAMXCNTL will then attempt to RETRY the shunted UOW(s) for the dataset.

Action: None

IAML1055 IAMXCNTL - UOWENQ found for UOW=(<UOWID>) for DSN(<dataset name>)

Reason: Informational message showing that the IAM CICS program IAMXCNTL has determined that there is at least one UOWENQ pending for the specified dataset. IAMXCNTL will then attempt to RETRY the shunted UOW(s) for the dataset.

Action: None
IAML1060  IAMXCNTL - No Enhanced-mode IAM files remain OPEN in this CICS region.

Reason: Informational message showing that the IAM CICS program IAMCNTL has determined that there are no more IAM files that are still OPEN to an IAMRLS region that was found to have terminated.

Action: None

IAML1061  IAMXCNTL - CLOSE requested for FILE(<fileid>) with DSN(<dataset name>) at hh:mm:ss

Reason: Informational message showing that the IAM CICS program IAMCNTL is requesting CICS to CLOSE the specified FILE associated with the specified dataset.

Action: None

IAML1062  IAMXCNTL - CLOSE completed for FILE(<fileid>) with DSN(<dataset name>) at hh:mm:ss

Reason: Informational message showing that the IAM CICS program IAMCNTL has completed a CICS CLOSE of the specified FILE associated with the specified dataset.

Action: None

IAML1063  IAMXCNTL - FORCECLOSE issued for FILE(<fileid>) and DSN(<dataset name>) at hh:mm:ss

Reason: Informational message showing that the IAM CICS program IAMCNTL is issuing a CICS FORCECLOSE of the specified FILE associated with the specified dataset.

Action: None

IAML1064  IAMXCNTL - OPEN completed for FILE(<fileid>) with DSN(<dataset name>) at hh:mm:ss

Reason: Informational message showing that the IAM CICS program IAMCNTL has completed a CICS OPEN for the specified FILE associated with the specified dataset.

Action: None

IAML1065  IAMXCNTL - DATASET RETRY requested for DSN(<dataset name>) at hh:mm:ss

Reason: Informational message showing that the IAM CICS program IAMCNTL has completed a CICS RETRY of shunted UOW(s) for the specified dataset.

Action: None

IAML1066  IAMXCNTL - DATASET BACKOUT requested for DSN(<dataset name>) at hh:mm:ss

Reason: Informational message showing that the IAM CICS program IAMCNTL has completed a CICS BACKOUT of pending UOW(s) for the specified dataset.
**IAM/RLS and IAM/PLEX Messages**

**Action:** None

**IAML1067** IAMXCNTL - DATASET SET UNAVAILABLE for DSN(<dataset name>) at hh:mm:ss

**Reason:** Informational message showing that the IAM CICS program IAMCNTL has completed setting the specified dataset status as UNAVAILABLE.

**Action:** None

**IAML1068** IAMXCNTL - <errtype> Error on <reqtype> for <cmdtype>

where:  
- `<errtype>` = text string describing the error
- `<reqtype>` = the CICS SPI request that was issued
- `<cmdtype>` = any command or request options

**Reason:** A generic error message that occurs when CICS returns an unexpected response to the IAM program IAMXCNTL when it issued a CICS SPI request.

**Action:** Check for the associated IAML1069 message containing the response and error codes associated with the request.

**IAML1069** IAMXCNTL - Resp='xxxxxxxx' Resp2='xxxxxxxx' EIBRCODE='xxxxxxxx'

**Reason:** Diagnostic message showing that the response and the CICS EXEC Interface Block Return Codes when it executed an SPI function.

**Action:** Check in the CICS SPI Programming Reference, for the explanation of the CICS response (RESP=xxxxxxxx), the CICS Response2 (RESP2=xxxxxxxx) and the EIBRCODE values. IAM uses the exits to communicate with a connected IAMRLS (or IAM/PLEX) region for sharing IAM files. Without this IAM will not be able to fully function. Determine and correct the indicated underlying cause and then restart the CICS region.

**IAML2000** IXCJOIN RC=NNNNNNNN,RSN=NNNNNNN,GROUP=CCCCCCCC

**Reason:** Results of IXCJOIN from IAMRLS during startup.

**Return Codes:** RC=00000004 and RSN=00000004 indicates first member to join the RLSGROUP. RC=00000000 and RSN=00000000 indicates members have already joined the group.

**Action:** If any other return or reason codes issued, call Innovation for assistance. Otherwise no action needed.

**IAML2001** IAMRLS GETMAIN FOR XCF AREA STORAGE FAILED.

**Reason:** There was insufficient storage available to obtain the work area needed for XCF processing.

**Action:** Will be followed by message IAML2021 IAMRLS SYSPLEX INITIALIZATION FAILED, IAMRLS WILL TERMINATE. Check your REGION parameters for the a IAMRLS address space.
IAML2002 IAMRLS IXCJOIN FAILED.

Reason: The IXCJOIN issued by IAMRLS failed, IAMRLS is unable to complete SYSPLEX initialization.

Action: Will be followed by message IAML2021 IAMRLS SYSPLEX INITIALIZATION FAILED, IAMRLS WILL TERMINATE. Call Innovation for assistance.

IAML2003 IAMRLS GETMAIN OF LOCAL XCF WORKAREA FAILED.

Reason: There was insufficient storage available to obtain the local work area needed for XCF processing.

Action: Will be followed by message IAML2021 IAMRLS SYSPLEX INITIALIZATION FAILED, IAMRLS WILL TERMINATE. Check your REGION parameters for the a IAMRLS address space.

IAML2004 IAMRLS MAXIMUM MEMBERS IN THIS RLSGROUP HAVE BEEN EXCEEDED, NO MORE ROOM FOR MEMBER ENTRIES.

Reason: The limit of 40 members in an RLSGROUP has been exceeded.

Action: You cannot start another IAMRLS region with this RLSGROUP until one active member of the group is quiesced and leaves the group.

IAML2005 IAMXCFS1 HAS BEEN POSTED FOR RLSID=CCCC

Reason: The XCF Group User Exit has been notified of an event like a member joining or leaving a group, and has posted the IAMXCFS1 subtask to process the event and update the status of the group.

Action: None

IAML2006 IAMXCFS1 GETMAIN FAILED

Reason: There was insufficient storage available to obtain the work area needed for the IAMXCFS1 subtask.

Action: Quiesce the IAMRLS region and check your Region parameters for the IAMRLS address space. The subtask will not be available to process XCF status updates until you recycle with sufficient REGION for the work area to be obtained.

IAML2007 IAMXCFS1 HAS BEEN ATTACHED

Reason: The IAMXCFS1 subtask has been attached during IAMRLS startup.

Action: None.

IAML2008 IXCQUERY RC=NNNNNNNN,RSN=NNNNNNN

Reason: The IAMXCFS1 subtask has issued an IXCQUERY to obtain the status information for an RLSGROUP.
Return Codes: The return and reason codes should always be zero. If an RC=00000004 RSN=00000004 is returned, it indicates the number of member entries returned exceeded the length of a work area. The subtask will getmain a new larger area and reissue the IXCQUERY.

Action: For any return or reason codes other then mentioned above, call Innovation for assistance.

IAML2009 IAMXCFS1 HAS BEEN POSTED WHILE INIT IN PROGRESS

Reason: The IAMXCFS1 subtask was posted to do work during its initialization.

Action: None, the subtask will be posted again after initialization completes.

IAML2010 IAMXCFS1 MAXIMUM MEMBERS IN THIS RLSGROUP EXCEEDED

Reason: The limit of 40 members in an RLSGROUP has been exceeded.

Action: You cannot start another IAMRLS region with this RLSGROUP until one active member of the group is quiesced and leaves the group.

IAML2011 IAMXCFS1 GETMAIN FAILED FOR LARGER ANSWER AREA, ALL MEMBERS CANNOT BE PROCESSED.

Reason: The IAMXCFS1 subtask was unable to acquire enough storage to process all of the members in the RLSGROUP.

Action: Call Innovation for assistance. You may not have enough REGION available for this IAMRLS address space.

IAML2012 IAMRLS SYSPLEX INITIALIZATION FAILED, IAMRLS WILL TERMINATE.

Reason: IAMRLS was unable to complete initialization of the SYSPLEX environment.

Action: This message will be preceded by another message indicating the reason why initialization failed.

IAML2013 PARMLIB CONTAINS ERRORS, CANNOT CONTINUE

Reason: The parmlib processor returned an error condition during startup.

Action: Check the IAMRLS system log and RLSLOGDD for messages indicating which parms were in error.

IAML2014 OPTION TABLE TYPE INCORRECT, IAMRLS WILL TERMINATE

Reason: There is a mismatch between the global option table and the level of module IAMRLS.

Action: Contact Innovation for assistance.
IAM/RLS and IAM/PLEX Messages 80.05

IAML2015 LEVEL MISMATCH BETWEEN IAMPLEX AND IAMRLS, IAMPLEX WILL TERMINATE

Reason: There is a mismatch between the level of the IAMPLEX module and the level of module IAMRLS.

Action: Contact Innovation for assistance.

IAML2016 IAMPLEX MODULE MUST BE USED IF RLSGROUP IS PROVIDED, IAMRLS WILL TERMINATE

Reason: The procedure that was used to start this IAMRLS region has provided and RLSGROUP in its startup parms but is executing PGM=IAMRLS.

Action: To use IAM SYSPLEX services with an RLSGROUP, PGM=IAMPLEX must be used.

IAML2017 RLSGROUP MUST BE PROVIDED WHEN EXECUTING IAMPLEX, IAMPLEX WILL TERMINATE

Reason: The procedure that was used to start this IAMRLS region has used PGM=IAMPLEX indicating it will use SYSPELX services, but has not provided an RLSGROUP.

Action: When requesting SYSPLEX services by coding the PGM=IAMPLEX in the startup procedure, and RLSGROUP must be provided in the startup parmlib.

IAML2018 IAMPLEX CANNOT BE USED, NOT LICENSED FOR IAMPLEX SUPPORT. IAMPLEX WILL TERMINATE

Reason: The procedure that was used to start this IAMRLS region has used PGM=IAMPLEX indicating it will use SYSPELX services, but this site is not licensed for IAMPLEX.

Action: Change the procedure to use PGM=IAMRLS, or contact Innovation for assistance.

IAML2019 DATASET NAME TABLE ALREADY IN STORAGE, TABLE PROCESSING BYPASSED

Reason: Another IAMRLS region has already processed the dataset name table and this IAMRLS region will use the one already in storage.

Action: None. If a change is required for the dataset name table then you can use the F IAMRLSnn,CHANGEDSNT command.

IAML2020 SYSTEM STATUS UPDATE MISSING FROM rlsid REPY Y FOR CLEANUP OR W TO WAIT

Reason: The IAMPLEX XCF user exit has been notified that the system status update is missing from CPU cccc and RLSID rrrr. There could be a problem on that LPAR causing this interruption.
**IAM/RLS and IAM/PLEX Messages**

**Action:** Reply 'Y' if you would like the IAMPLEX region issuing the message to cleanup for the IAMPLEX region not responding. This will prevent messages from being sent that cannot be handled by the IAMPLEX region having the problem. If you know of a reason why the LPAR is not responding and expect to resolve the situation shortly, you can respond 'W' and IAMPLEX will wait for the next notification that the IAMPLEX region has started updating the system status again.

**IAML2300**  
IXCMSGI FAILED, RC=xx REAS=xxxxxxxx MSGID=xxxxxxxx R=rlsid T=rlsid

**Reason:** IAM/PLEX attempted to retrieve the XCF message data being sent, but encountered the indicated error. The routing (R=) and the target (T=) rlsids are provided.

**Action:** Take a dump of the IAM/PLEX address space that the error message occurred on, and contact Innovation technical support.

**IAML2302**  
INSUFFICIENT STORAGE FOR ANOTHER OCQ POOL

**Reason:** The target IAM/PLEX address space was unable to obtain the necessary storage to process an OPEN or a job step termination request. The target IAM/PLEX will attempt to send a message back to the router IAM/PLEX to notify it of the problem.

**Action:** Take a dump of the IAM/PLEX address space that the error message occurred on, and contact Innovation technical support.

**IAML2303**  
MSG IGNORED, INVALID MSG TYPE OF ###

**Reason:** An IAM/PLEX address space received an invalid message from XCF. The message will be ignored.

**Action:** Take a dump of the IAM/PLEX address space that the error message occurred on, and contact Innovation technical support.

**IAML2304**  
TARGET OCQ VALIDATION FAILED, OCQ @ xxxxxxxx

**Reason:** The target IAM/PLEX address space encountered a validation error on a close request. The file may have already been closed. The target IAM/PLEX will attempt to send a message back to the router IAM/PLEX to notify it of the problem.

**Action:** Take a dump of the IAM/PLEX address space that the error message occurred on, and contact Innovation technical support.

**IAML2305**  
ROUTER OCQ VALIDATION FAILED, OCQ @ xxxxxxxx BAD OCQ ADDRESS
TOKEN MISMATCH
OCQ TOKEN MISMATCH
MESSAGE ID MISMATCH

**Reason:** The router IAM/PLEX address space encountered a failure in validating a response to a close or job termination request.

**Action:** Take a dump of the IAM/PLEX address space that the error message occurred on, and contact Innovation technical support.
IAML2306  
**REQ FOR MSG STORAGE FAILED, RC=xx LENGTH = n MSGID=xxxxxxxx R=rlsid T=rlsid**

**Reason:** The IAM/PLEX message processing exit was unable to obtain storage to process a message from XCF.

**Action:** Take a dump of the IAM/PLEX address space that the error message occurred on, and contact Innovation technical support.

IAML2307  
**REQUIRED MSG DATA NOT SENT, MSGID=xxxxxxxx R=rlsid T=rlsid**

**Reason:** The IAM/PLEX message processing exit received an invalid message.

**Action:** Take a dump of the IAM/PLEX address space that the error message occurred on, and contact Innovation technical support.

IAML2308  
**UNEXPECTED XPL TYPE / LENGTH, MSGID=xxxxxxxx R=xxxx T=xxxx**

**Reason:** An IAM/PLEX address space received an invalid message from XCF. The message will be ignored.

**Action:** Take a dump of the IAM/PLEX address space that the error message occurred on, and contact Innovation technical support.

IAML2312  
**INSUFFICIENT STORAGE FOR ANOTHER IOQ POOL**

**Reason:** The target IAM/PLEX address space was unable to obtain the necessary storage to process an I/O or a syncpoint request. The target IAM/PLEX will attempt to send a message back to the router IAM/PLEX to notify it of the problem which will result in a failing return code for the request.

**Action:** Take a dump of the IAM/PLEX address space that the error message occurred on, and contact Innovation technical support.

IAML2314  
**TARGET IOQ ALREADY IN USE, IOQ @ xxxxxxxx MSGID=xxxxxxxx**

**Reason:** The target IAM/PLEX address space received an I/O request for an IOQ (RPL) that was already busy processing an I/O request. The router will be notified of the error and result in a failing return code on the originator of the I/O request.

**Action:** Take a dump of the IAM/PLEX address space that the error message occurred on, and contact Innovation technical support.

IAML2315  
**OPEN MSGID:xxxxxxxx DOES NOT MATCH OCQ MSGID: xxxxxxxx OCQ @ xxxxxxxx HAS INVALID OCQ ADDRESS: xxxxxxxxx**

**Reason:** The router received a response for an active open request however the response has incorrect or conflicting information.

**Action:** Take a dump of the IAM/PLEX address space that the error message occurred on, and contact Innovation technical support.
IAML2316  OPEN RESPONSE RECEIVED FOR OCQ @ xxxxxxxx FOR JOB jobname
INDICATORS INVALID

Reason: The router received a response for an active open request however the response has incorrect or conflicting information.

Action: Take a dump of the IAM/PLEX address space that the error message occurred on, and contact Innovation technical support.

IAML2317  I/O MSGID: xxxxxxxx DOES NOT MATCH IOQ MSGID: xxxxxxxx IOQ @ xxxxxxxx
IOQ @ xxxxxxxx INDICATORS INVALID
HAS INVALID IOQ ADDRESS: xxxxxxxx
SRB SCHEDULE FAILED, RC= xxxxxxxx

Reason: The router received a response for an active open request however the response has incorrect or conflicting information. If the error is an SRB Schedule failed then the response data was good however IAM/PLEX did not successfully notify the originator’s address space of the I/O completion.

Action: If the failure is an SRB Schedule failure and the originating job is no longer active, then no action is necessary. Otherwise take a dump of the IAM/PLEX address space that the error message occurred on, and contact Innovation technical support.

IAML2318  BMON MSGID: xxxxxxxx DOES NOT MATCH IOQ MSGID: xxxxxxxx IOQ @ xxxxxxxx
HAS INVALID ICQ ADDRESS: xxxxxxxx

Reason: The router received a response for an active syncpoint request however the response has incorrect or conflicting information.

Action: Take a dump of the IAM/PLEX address space that the error message occurred on, and contact Innovation technical support.

IAML2319  BSYN MSGID: xxxxxxxx DOES NOT MATCH IOQ MSGID: xxxxxxxx IOQ @ xxxxxxxx
HAS INVALID IOQ ADDRESS: xxxxxxxx

Reason: The router received a response for an active syncpoint request however the response has incorrect or conflicting information.

Action: Take a dump of the IAM/PLEX address space that the error message occurred on, and contact Innovation technical support.

IAML2321  MSGU PROCESSING OPEN REQUEST FOR MSGID=xxxxxxxx R=xxxx T=xxxx

Reason: Message on a TARGET IAM/PLEX address space indicating it has received a request to OPEN an IAM file, and is proceeding with validating and scheduling the request.

Action: None, and informational only message that will only be issued when the MESSAGES=ALL parameter is specified.
IAML2322 MSGU COMPLETED OPEN REQUEST FOR MSGID=xxxxxxxx R=xxxx T=xxxx

Reason: Message on a TARGET IAM/PLEX address space indicating it has received a request to OPEN an IAM file. The request has been validated and scheduled for processing by the Open/Close task.

Action: None, and informational only message that will only be issued when the MESSAGES=ALL parameter is specified.

IAML2323 MSGU PROCESSING CLOSE REQUEST FOR MSGID=xxxxxxxx R=xxxx T=xxxx

Reason: Message on a TARGET IAM/PLEX address space indicating it has received a request to CLOSE an IAM file, and is proceeding with validating and scheduling the request.

Action: None, and informational only message that will only be issued when the MESSAGES=ALL parameter is specified.

IAML2324 MSGU COMPLETED CLOSE REQUEST FOR MSGID=xxxxxxxx R=xxxx T=xxxx

Reason: Message on a TARGET IAM/PLEX address space indicating it has received a request to CLOSE an IAM file. The request has been validated and scheduled for processing by the Open/Close task.

Action: None, and informational only message that will only be issued when the MESSAGES=ALL parameter is specified.

IAML2325 MSGU PROCESSING BMON REQUEST FOR MSGID=xxxxxxxx R=xxxx T=xxxx

Reason: Message on a TARGET IAM/PLEX address space indicating it has received a BMON request and is proceeding with validating and scheduling the request.

Action: None, and informational only message that will only be issued when the MESSAGES=ALL parameter is specified.

IAML2326 MSGU COMPLETED BMON REQUEST FOR MSGID=xxxxxxxx R=xxxx T=xxxx

Reason: Message on a TARGET IAM/PLEX address space indicating it has received a BMON request. The request has been validated and scheduled for processing.

Action: None, and informational only message that will only be issued when the MESSAGES=ALL parameter is specified.

IAML2327 MSGU PROCESSING BSYN REQUEST FOR MSGID=xxxxxxxx R=xxxx T=xxxx

Reason: Message on a TARGET IAM/PLEX address space indicating it has received a BSYN (syncpoint) request and is proceeding with validating and scheduling the request.

Action: None, and informational only message that will only be issued when the MESSAGES=ALL parameter is specified.
IAML2328  MSGU COMPLETED BSYN REQUEST FOR MSGID=xxxxxxxx R=xxxx T=xxxx

**Reason:** Message on a TARGET IAM/PLEX address space indicating it has received a BSYN (syncpoint) request. The request has been validated and scheduled for processing.

**Action:** None, and informational only message that will only be issued when the MESSAGES=ALL parameter is specified.

IAML2329  FUNCTION RELEASE FAILED, RC=xxxx

**Reason:** A call to the release service to activate a task in the target IAMPLEX address space failed with the indicated return code. The request may not be processed.

**Action:** Take a dump of the IAMPLEX address space on which the message occurred and contact Innovation for support.

IAML2400  RCWA TRM RELEASE FAILED, RC=xxxx

**Reason:** A call to terminate the Remote Open Close subtask failed.

**Action:** Contact Innovation for support.

IAML2401  OCWA TRM RELEASE FAILED, RC=xxxx

**Reason:** A call to terminate the Open Close subtask failed.

**Action:** Contact Innovation for support.

IAML2402  OCWA CLO RELEASE FAILED, RC=xxxx

**Reason:** A call to the Close Subtask to close a file failed.

**Action:** Contact Innovation for support.

IAML2403  IOWA TRM RELEASE FAILED, RC=xxxx

**Reason:** A call to terminate an I/O subtask failed.

**Action:** Contact Innovation for support.

IAML2404  IOWA I/O RELEASE FAILED, RC=xxxx

**Reason:** A call to release an I/O subtask from the idle queue failed.

**Action:** Contact Innovation for support.
IAML3000  cccccc SELECT ENTRIES AND ppppp PRIOR ENTRIES EXCEEDS MAXIMUM OF mmmmmm

Reason: The maximum total number of SELECTDSN table entries has been exceeded.

Action: There is currently a 5000 dsn limit for the SELECT and EXCLUDE tables, if you need more than that call Innovation for support.

IAML3001  TABLE BUILD FAILED/NOT ATTEMPTED -- VALUES ARE FROM PRIOR TABLES

Reason: The table failed with one or more preceding error messages.

Action: Correct the identified errors and rerun the CHANGEDSNT command.

IAML3002  MTKSELTB VALUE IS ZERO (0) -- NO TABLE LISTED

Reason: There is no existing SELECT DSN table in storage.

Action: None required. If there should be a SELECT table, take a dump of the IAM/RLS or IAM/PLEX region and contact Innovation for support.

IAML3003  MTKEXCTB VALUE IS ZERO (0) -- NO TABLE LISTED

Reason: There is no existing EXCLUDE DSN table in storage.

Action: None required. If there should be an EXCLUDE table, take a dump of the IAM/RLS or IAM/PLEX region and contact Innovation for support.

IAML3004  ERRORS IN ....... MASKS RESULTS IN AN EMPTY TABLE -- JOB TERMINATED

Reason: Due to prior error in the input control information identified in one or more prior messages, IAM was unable to create the DSN table.

Action: Correct the identified errors and rerun the CHANGEDSNT command.

IAML3005  SYNTAX ERROR -- INVALID CHARACTERS

Reason: A syntax error due to invalid characters was encountered when reading in the dataset name table.

Action: Correct the error in the table and retry.

IAML3006  SYNTAX ERROR -- LAST CHARACTER CANNOT BE PERIOD(.)

Reason: A syntax error due to the last characters being a period.

Action: Correct the error in the table and retry.

IAML3007  SYNTAX ERROR -- DATA REDUCES TO NO SEARCH ARGUEMENT

Reason: A syntax error due to data reducing to no search argument.

Action: Correct the error in the table and retry.
IAML3008  SYNTAX ERROR -- QUALIFIER LONGER THAN EIGHT (8) BYTES

Reason:  A syntax error due a qualifier being longer than 8 bytes.

Action:  Correct the error in the table and retry.

IAML3009  SYNTAX ERROR -- MASK CHARACTER(X) INVALID FOLLOWING ASTERISK(*)

Reason:  A syntax error due to a mask character of X following an asterisk.

Action:  Correct the error in the table and retry.

IAML3010  SYNTAX ERROR -- MORE THAN TWO(2) CONTIGUOUS ASTERISKS(*)

Reason:  A syntax error due to two contiguous asterisks being found.

Action:  Correct the error in the table and retry.

IAML3011  SYNTAX ERROR -- MORE THAN TWENTY-ONE (21) INDEX LEVELS

Reason:  A syntax error due to more than twenty-one index levels found.

Action:  Correct the error in the table and retry.

IAML3012  ERROR IN MASK: ..........0.........0.........0.........0...4

Reason:  An error in a mask was found.

Action:  Correct the error in the table and retry.

IAML9000  IAMZAPR MAINTENANCE APPLIED

Reason:  The pfts supplied as input to the APPLY command have been successfully applied to the virtual storage copies of the IAMRLS processing modules.

Action:  None

IAML9001  IAMZAPR MAINTENANCE RESTORED

Reason:  The ptf supplied as input to the RESTORE command have been successfully removed from the virtual copies of the IAMRLS processing modules.

Action:  None

IAML9002  IAMZAPR MAINTENANCE NOT APPLIED

Reason:  The ptf supplied as input to the APPLY command have not been successfully applied to the virtual storage copies of the IAMRLS processing modules.

Action:  Browse the output from SYSPRINT the DD and RLSLOGDD for additional information as to why the ptf was not applied.
IAML9003  IAMZAPR MAINTENANCE NOT RESTORED

**Reason:** The ptf supplied as input to the RESTORE command have not been successfully removed from the virtual storage copies of the IAMRLS processing modules.

**Action:** Browse the output SYSPRINT DD and RLSLOGDD for additional information as to why the ptf was not removed.

IAML9004  IAMZAPR ALLOCATION FAILED FOR DSN - dsname

**Reason:** The allocation of the dataset specified on either the apply or restore command has failed.

**Action:** Check the syntax of the command entered and make sure the dataset name is correct and is available. Re-enter the command with the correct dataset name.

IAML9005  IAMZAPR OPEN FAILED FOR DSN - dsname

**Reason:** The open of the dataset specified on either the apply or restore command has failed.

**Action:** Check the syntax of the command entered and make sure the dataset name is correct and is available. Re-enter the command with the correct dataset name.

IAML9006  IAMZAPR OPEN OF SYSPRINT DD FAILED

**Reason:** IAMZAPR was unable to open the SYSPRINT DD during APPLY or RESTORE command processing.

**Action:** Make sure the IAMRLS JCL used to start the IAM RLS address space has a SYSPRINT DD included. If it does, browse the RLSLOGDD, JESLOG, or SYSMSGS for additional messages documenting the reason for the open failure.

IAML9007  IAMZAPR OPEN OF SYSLIB DD FAILED

**Reason:** IAMZAPR was unable to open the SYSLIB DD during APPLY or RESTORE command processing.

**Action:** Make sure the IAMRLS JCL used to start the IAM RLS address space has a SYSLIB DD included. If it does, browse the RLSLOGDD, JESLOG, or SYSMSGS for additional messages documenting the reason for the open failure.

IAML9008  IAMZAPR VIFKEY MISMATCH BETWEEN IAMZAPR AND IAMVECTB FROM SYSLIB

**Reason:** There is a module level mismatch between the execution load library and the SYSLIB load library included in the startup JCL for IAMRLS.

**Action:** Make sure the SYSLIB DD points to the current IAM load library and is the same as the IAM load library in use from the LINKLIST/JOBLIB/STEPLIB.
IAML9009  IAMZAPR VECTOR TABLE FOR CURRENT LEVEL NOT FOUND, VIF NOT ACTIVE

Reason: IAMZAPR was unable to locate the IAM VECTOR TABLE for the current level of the VSAM INTERFACE that matches the level of the IAMZAPR module.

Action: Browse the RLSLOGDD and the SYSTEM LOG to determine why VIF was not started, or if it was stopped.

IAML9010  IAMZAPR PROCESSING PTF nnnnnnnnn

Reason: The PTF being processed by either the APPLY or RESTORE command is documented here.

Action: None, informational message only.

IAML9011  THE ABOVE STATEMENT IS MISSING AN OPERAND. REASON=9

Reason: IAMZAPR has encountered a missing operand in the PTF control statements.

Action: Correct the error with the PTF in the input library supplied to either the APPLY or RESTORE command and re-enter the command.

IAML9012  THE ABOVE STATEMENT CONTAINS AN ODD NUMBER OF DIGITS. REASON=15

Reason: The PTF input contains an odd number of digits in one of the offsets or data portions of the zap.

Action: Correct the error with the PTF in the input library supplied to either the APPLY or RESTORE command and re-enter the command.

IAML9013  THE ABOVE STATEMENT CONTAINS AN INVALID HEXADECIMAL DIGIT. REASON=16

Reason: The PTF input contains an invalid hexadecimal digit in one of the offsets or data portions of the zap.

Action: Correct the error with the PTF in the input library supplied to either the APPLY or RESTORE command and re-enter the command.

IAML9014  MODULE IAMVECTB NOT FOUND. REASON=19

Reason: The IAMVECTB module could not be found in LPA. The operation is failed.

Action: Browse the SYSLOG to see why VIF is not longer active.

IAML9015  THE ABOVE MODULE IS NOT ACTIVE. REASON=7

Reason: The module that was named on the PTF NAME statement was found in storage, but is marked as inactive.

Action: VIF must have been stopped, in order to apply the PTF’s to the virtual storage copy of the module you will have to restart VIF.
IAML9016  THE ABOVE MODULE WAS NOT FOUND. REASON=8

Reason:   The module that was named on the PTF NAME statement was either not in
            the table of modules that can be zapped, or could not be loaded or found in storage.

Action:   Call Innovation for help in diagnosing this problem.

IAML9017  NAME CARD MISSING. REASON=13

Reason:   The PTF supplied as input to the APPLY or RESTORE command is missing
            a NAME statement.

Action:   Check your input against the PTF that was sent to you and if both are missing
            the NAME statement, call Innovation for a corrected PTF.

IAML9018  THE ABOVE STATEMENT OFFSET IS INVALID. REASON=11

Reason:   The offset in the PTF provided is invalid.

Action:   Check your input against the PTF that was sent to you and if both are the
            same, call Innovation for a corrected PTF.

IAML9019  THE MODULE WAS FOUND IN PLPA OR IN READ ONLY NUCLEUS WHICH IS NOT
            VALID. REASON=11

Reason:   The module named on the NAME statement in the PTF supplied as input to
            either the APPLY or RESTORE command was found in PLPA or the read only nucleus.

Action:   None of the IAM modules should be in PLPA or the read only nucleus. If you
            have put IAM modules in PLPA then you cannot use the online zap command to apply
            maintenance against them, you must use superzap and IPL with a CLPA.

IAML9020  THE ABOVE STATEMENT OFFSET IS NOT WITHIN THE MODULE. REASON=12

Reason:   The offset in the PTF provided is invalid.

Action:   Check your input against the PTF that was sent to you and if both are the
            same, call Innovation for a corrected PTF.

IAML9021  MODULE EP ADDRESS  = xxxxxxxx

Reason:   The entry point of the module about to be zapped is documented here.

Action:   None

IAML9022  VERIFY FAILED DATA = xxxxxxxx

Reason:   The PTF failed to verify with the data printed in this message.

Action:   Check your input against the PTF that was sent to you and if both are the
            same, call Innovation for a corrected PTF.
IAML9023  OLD DATA TO REPLACE= xxxxxxxx

**Reason:** The data being replaced in the module by the PTF is documented here.

**Action:** None

IAML9024  BASE CARD IS NOT SUPPORTED FOR LPA. REASON=14

**Reason:** There is a BASE card included with the PTF, which is not allowed for the intercept modules located in LPA.

**Action:** This PTF must be applied using superzap, and a recycle of IAMRLS and VIF scheduled to pick up the maintenance.

IAML9025  SETSSI DOES NOT CONTAIN 8 HEXADECIMAL DIGITS. REASON=24

**Reason:** There is an invalid number of digits in a SETSSI keyword in the PTF.

**Action:** Check your input against the PTF that was sent to you and if both are the same, call Innovation for a corrected PTF.

IAML9026  CHECKSUM IS xxxxxxxx

**Reason:** The checksum data in the PTF is documented here.

**Action:** None, informational message only.

IAML9027  THE CURRENT CHECKSUM VALUE IS.................xxxxxxxxxx

**Reason:** The actual checksum of the zap statements is documented here.

**Action:** None

IAML9029  IAMZAPR WILL APPLY ZAPS TO LPA NOW

**Reason:** The checksum and verify data is correct, the zap will be applied to the virtual storage copy of the module.

**Action:** None

IAML9031  CHECKSUM IS CORRECT

**Reason:** The checksum data supplied in the PTF is correct.

**Action:** None

IAML9032  CHECKSUM IS WRONG - PLEASE CORRECT THE ZAP. REASON=1

**Reason:** The checksum data provided in the PTF is incorrect.

**Action:** Check your input against the PTF that was sent to you and if both are the same, call Innovation for a corrected PTF.
IAML9033  IF YOU DO NOT FIND THE ERROR, PLEASE CALL INNOVATION TECH SUPPORT (KEEP THE OUTPUT LISTING).

Reason:  This comes out after the IAML9033 message to remind you call Innovation if you cannot find the error.

Action:  Call Innovation if you cannot find the error.

IAML9034  THE FOLLOWING PAGES CONTAIN THE PARTIAL CHECKSUM OF EVERY STATEMENT (FOR INNOVATION USE ONLY)

Reason:  Documentation of the error in applying the PTF.

Action:  Save for Innovation Tech Support.

IAML9035  THE ABOVE STATEMENT IS INVALID.  REASON=10

Reason:  The PTF statement printed above this message is incorrect.

Action:  Check your input against the PTF that was sent to you and if both are the same, call Innovation for a corrected PTF.

IAML9100   'name' REGISTERED FOR ARM PROCESSING

Reason:  The IAMRLS or IAM/PLEX region named in the message has registered for ARM processing with the displayed return and reason codes.

Action:  If the return code is zero or four then no action necessary, If higher then 4 contact Innovation for support.

IAML9101   'name' READY FOR ARM PROCESSING

Reason:  The IAMRLS or IAM/PLEX region named in the message is now ready for ARM processing with the displayed return and reason codes.

Action:  If the return code is zero or four then no action necessary If higher then 4 contact Innovation for support.

IAML9102   'name' DEREGISTERED FOR ARM PROCESSING

Reason:  The IAMRLS or IAM/PLEX region named in the message has deregistered for ARM processing with the displayed return and reason codes.

Action:  If the return code is zero or four then no action necessary If higher then 4 contact Innovation for support.
80.10 IAM ABEND Codes

As a general rule, the IAM access method avoids intentionally abending, but rather passes return codes and error codes back to the calling program. Most of the abend codes listed below are for the various utility programs available with IAM, including IAMRECVR, IAMZAPOP, IAMSMFVS, and IAMSMF.

The abend codes issued by the old IAM Native and ISAM interfaces have been removed from the manual. They are documented in the ICL library that was loaded as part of the product installation. Refer to member OLDABEND.

<table>
<thead>
<tr>
<th>Abend Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U0102</td>
<td>The UPAD exit routine returned to IAM with the contents of R1 being either x'00', or not pointing to the parameter area that was passed from IAM to the UPAD exit.</td>
</tr>
<tr>
<td>U0183</td>
<td>An error condition occurred during file load processing. An internal error code is placed in R15. Contact Innovation for support and assistance.</td>
</tr>
<tr>
<td>U0184</td>
<td>An error condition occurred during IAM processing, and the user had supplied a //IAMDEBUG DD DUMMY DD statement. For various errors, IAM will check if an IAMDEBUG DD has been supplied, and rather than returning an error code and non-zero return code, IAM will issue an abend. The primary purpose of this capability is to add in problem diagnosis.</td>
</tr>
<tr>
<td>U0185</td>
<td>IAM-RLS was unable to obtain the requested record lock, and the RLS LOCK parameter indicated that it should abend the job. R15 has the associated logical error code. See Section 80.21 for the meaning of the code in R15.</td>
</tr>
<tr>
<td>U0402</td>
<td>An IAM utility program encountered a failure attempting to open the required SYSPRINT DD statement. Most likely the DD statement is either missing, or incorrectly spelled.</td>
</tr>
<tr>
<td>U0502</td>
<td>An IAM utility program encountered an error processing the required control card input. The required SYSIN DD statement is missing or incorrectly spelled, or no control cards were supplied. There should be an error message provided on the SYSPRINT listing indicating the cause of the problem.</td>
</tr>
<tr>
<td>U0600</td>
<td>An IAM utility program encountered an error processing an input or output file. Refer to SYSPRINT for messages detailing the error encountered.</td>
</tr>
<tr>
<td>U0658</td>
<td>An IAM utility program encountered a module that did not match the level of the utility program being executed. This may indicate an incorrect STEPLIB, or a problem with the product installation.</td>
</tr>
<tr>
<td>U0659</td>
<td>An IAM utility program encountered an unexpected logical error during processing. There should be messages on SYSPRINT further explaining the cause of the problem.</td>
</tr>
<tr>
<td>U0660</td>
<td>An IAM utility program encountered an internal save area stack overflow. Contact Innovation for assistance.</td>
</tr>
<tr>
<td>U0777</td>
<td>IAMSTART encountered an error with the SETLOCK service. Contact Innovation for assistance.</td>
</tr>
<tr>
<td>U0900</td>
<td>The IAM utility IAMEXTND, encountered a correctable user error. Refer to the SYSPRINT output for messages detailing the cause of the error. Contact Innovation.</td>
</tr>
<tr>
<td>Abend Code</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>U0901</td>
<td>The IAM utility IAMEXTND encountered a serious error. Refer to the SYSPRINT listing for messages indicating the reason for the failure. Contact Innovation.</td>
</tr>
<tr>
<td>U0902</td>
<td>The IAM utility IAMEXTND encountered a serious processing error. Refer to the SYSPRINT listing for error messages. Contact Innovation.</td>
</tr>
</tbody>
</table>
80.20 IAM Catalog Return Codes

The following is a list of return codes and reason codes that IAM will set for a file DEFINE. When a file is defined under IDCAMS, these codes appear in the IDC3009I message. IDCAMS may print out other messages that relate to the failure code. Whenever possible, IAM uses codes that will have the same or similar meaning for VSAM files, however that is not always possible. There will be an IAM error message, with the IAMW prefix that should identify the error in more detail. Also, for most allocations, there will also be error messages generated by the failed dynamic allocation request, which IAM will print out. Due to the way IDCAMS displays messages, the error messages printed by IAM will actually appear before the card images for the actual DEFINE. Then, IDCAMS will print out its own error messages based upon the codes that IAM set.

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Reason Code</th>
<th>Error description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>6</td>
<td>Invalid type specified on a DELETE request.</td>
</tr>
<tr>
<td>8</td>
<td>38</td>
<td>Duplicate dataset name found in the catalog, or on the volume(s) to which the dataset is being defined.</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>SMS has failed the allocation request. Refer to the associated SMS error messages for additional information on the exact cause of the error.</td>
</tr>
<tr>
<td>22</td>
<td>8</td>
<td>The user attempted to define a cluster of a type not supported by IAM. IAM does not support LINEAR type of VSAM Clusters.</td>
</tr>
<tr>
<td>22</td>
<td>12</td>
<td>The user attempted to define an IAM AIX, PATH, or RRDS type of dataset. The version of IAM being used is not licensed for the type of file that the user is attempting to define.</td>
</tr>
<tr>
<td>42</td>
<td>nnnn</td>
<td>MVS/ESA DADSM allocation of the dataset failed. The reason code is the return code from DADSM.</td>
</tr>
<tr>
<td>44</td>
<td>12</td>
<td>The work area provided by the caller of a request for information from the catalog for an IAM file was not large enough to contain all of the requested information.</td>
</tr>
<tr>
<td>54</td>
<td>nnnn</td>
<td>MVS/XA DADSM allocation of the dataset failed. The reason code is the return code from DADSM.</td>
</tr>
<tr>
<td>56</td>
<td>6</td>
<td>User is not authorized to define the dataset, according to the security system.</td>
</tr>
<tr>
<td>58</td>
<td>nnnn</td>
<td>On a DEFINE RECATALOG request, the attempt to OBTAIN the VTOC information for the specified dataset failed. The reason code is the return code from the OBTAIN service.</td>
</tr>
<tr>
<td>58</td>
<td>4</td>
<td>A CVAF service request issued by IAM during the define of an IAM file failed. CVAF indicated that the volume on which the dataset was defined was not mounted. There should be a corresponding IAMW50 error message.</td>
</tr>
<tr>
<td>58</td>
<td>8</td>
<td>A CVAF service request issued by IAM during the define process of an IAM file failed. CVAF indicated that the DSCB for the IAM file was not found on the volume to which the IAM file was defined. There should be a corresponding IAMW50 error message.</td>
</tr>
<tr>
<td>58</td>
<td>12</td>
<td>A CVAF service request issued by IAM during the define process of an IAM file failed. The CVAF return code and reason code are on the associated IAMW50 error message.</td>
</tr>
<tr>
<td>60</td>
<td>4</td>
<td>A catalog information request was issued (locate SVC) which appeared to be for an IAM file, however either an error occurred during IAM processing, or the file is not an IAM file. Normally, this return code will only be set if an /*IAMDEBUG DD DUMMY statement is included in the JCL.</td>
</tr>
<tr>
<td>62</td>
<td>0</td>
<td>The initialization of the IAM file being defined failed. There should be IAMWxx messages indicating the cause of the error.</td>
</tr>
<tr>
<td>Return Code</td>
<td>Reason Code</td>
<td>Error description:</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>62</td>
<td>84</td>
<td>IAM found an error within the block of the file containing the alternate index and path information.</td>
</tr>
<tr>
<td>62</td>
<td>88</td>
<td>There was insufficient space to add additional alternate index or path information. Using a larger block size for the base cluster may correct the problem.</td>
</tr>
<tr>
<td>68</td>
<td>20</td>
<td>The define request failed because there was insufficient space on the specified volume(s) to contain the dataset.</td>
</tr>
<tr>
<td>72</td>
<td>4</td>
<td>During a define of an IAM file, IAM was not able to find one or more of the requested volumes online.</td>
</tr>
<tr>
<td>80</td>
<td>2</td>
<td>Attempted to define an AIX for an RRDS type of dataset. As per VSAM restrictions, AIX’s cannot be used for RRDS type of datasets.</td>
</tr>
<tr>
<td>80</td>
<td>4</td>
<td>During a define of an alternate index or path, the related dataset or path entry dataset was not found. Make sure that the related dataset name or path entry dataset name is specified correctly.</td>
</tr>
<tr>
<td>80</td>
<td>10</td>
<td>A define of an alternate index or path was missing the related or path entry name.</td>
</tr>
<tr>
<td>80</td>
<td>12</td>
<td>During a define of an alternate index or path, the related dataset or path entry specified was invalid for the type of dataset being defined. Make sure that the related dataset is an IAM dataset of the appropriate type (base for an alternate index define, and base or alternate index for a path define.)</td>
</tr>
<tr>
<td>86</td>
<td>4</td>
<td>During a define recatalog of an IAM dataset, IAM encountered a failure attempting to determine the attributes of the file. The file most likely is not an IAM file.</td>
</tr>
<tr>
<td>86</td>
<td>6</td>
<td>During a define recatalog of an IAM dataset, IAM could not find the dataset on the specified volume.</td>
</tr>
<tr>
<td>96</td>
<td>4</td>
<td>Maximum record size for dataset exceed IAM’s maximum record size.</td>
</tr>
<tr>
<td>96</td>
<td>8</td>
<td>The base cluster record size is not large enough to contain the specified alternate key, or the calculated minimum alternate index record size exceeds the record size defined for the alternate index.</td>
</tr>
<tr>
<td>132</td>
<td>xx</td>
<td>During the define of an IAM file, the internal parameter list usually generated by IDCAMS was missing data. The reason code indicates the particular field that was not provided. This most likely is not a user error, but rather an error on the part of the software issuing the define.</td>
</tr>
<tr>
<td>132</td>
<td>2</td>
<td>No VOLUME FVT was found in the parameter list.</td>
</tr>
<tr>
<td>132</td>
<td>4</td>
<td>No AMDSB FVT was found in the parameter list.</td>
</tr>
<tr>
<td>132</td>
<td>8</td>
<td>No Average LRECL FVT was found in the parameter list.</td>
</tr>
<tr>
<td>132</td>
<td>10</td>
<td>No Space FVT was found in the parameter list.</td>
</tr>
<tr>
<td>132</td>
<td>26</td>
<td>No SPACE was found the FPL .</td>
</tr>
<tr>
<td>132</td>
<td>34</td>
<td>No attribute FPL was found (RGATTR FPL) on the define of an IAM Path or alternate index.</td>
</tr>
<tr>
<td>132</td>
<td>48</td>
<td>No AMDSB in the FPL.</td>
</tr>
<tr>
<td>136</td>
<td>2</td>
<td>No VOLUME information length.</td>
</tr>
<tr>
<td>136</td>
<td>6</td>
<td>No CLUSTER FVT found.</td>
</tr>
<tr>
<td>136</td>
<td>18</td>
<td>No Average LRECL found in the FPL</td>
</tr>
<tr>
<td>140</td>
<td>36</td>
<td>The IAMOVRID override statements contained an error, either in syntax or an unknown keyword was specified.</td>
</tr>
<tr>
<td>Return Code</td>
<td>Reason Code</td>
<td>Error description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>140</td>
<td>72</td>
<td>Duplicate volume found in volume list.</td>
</tr>
<tr>
<td>140</td>
<td>120</td>
<td>The define of an IAM file failed because it was assigned to a DATACLASS that specified COMPACTION (SMS Compression). IAM provides compression services so the SMS Compression can not be utilized.</td>
</tr>
<tr>
<td>168</td>
<td>2</td>
<td>IAM's internal device characteristics table does not contain an entry for the type of device on which the user is attempting to define an IAM file. If the device is a valid DASD device, contact Innovation for support.</td>
</tr>
<tr>
<td>176</td>
<td>0</td>
<td>The define of an IAM file failed because there was insufficient room in the VTOC on the specified volume(s) to contain an entry for the new file. The dataset will have to be allocated on a different volume, or some datasets will need to be deleted from the target volume. Another option is to enlarge the size of the VTOC.</td>
</tr>
<tr>
<td>184</td>
<td>4</td>
<td>The define of an IAM file failed because the dataset was in use by another job.</td>
</tr>
<tr>
<td>192</td>
<td>0</td>
<td>During the file define, IAM determined that the maximum record length being requested is longer than IAM supports for files without the SPANNED attribute. IAM can support records up to 32,760 bytes long if they are not spanned.</td>
</tr>
<tr>
<td>204</td>
<td>0</td>
<td>The key length plus the key offset exceeds the specified record length.</td>
</tr>
<tr>
<td>240</td>
<td>4</td>
<td>IAM was unable to determine the device characteristics for the volume requested on the file define request.</td>
</tr>
<tr>
<td>240</td>
<td>36</td>
<td>On the file define, the required DD statement for the volume(s) on which the dataset is to be defined was not found, or was not provided. This return code should normally not be seen by an end user.</td>
</tr>
</tbody>
</table>
IAM adheres to the VSAM application programming interface guidelines. When a request fails, IAM sets a non-zero return code in register 15, and provides a reason code within the RPL. Abends are avoided as much as possible. The appropriate exit routine, EODAD, LERAD, or SYNAD will be given control if so specified by the application program. It is the responsibility of the application program to verify the results of each I/O request, and take the action it deems appropriate in response to any error circumstance. As a result of a failing request, IAM will set the return code and error code to match the VSAM codes as much as is possible.

The table below indicates the return code, which is returned in register 15, and the error code, which is returned in the RPL field RPLERRCD. (The return code is also in the RPL, in field RPLRTNCD.) The error code is returned to an application program through the use of the SHOWCB macro, by requesting the FDBK field of an RPL.

<table>
<thead>
<tr>
<th>Return Code (Decimal)</th>
<th>Error Code (Hex)</th>
<th>Description of Error Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X'00'</td>
<td>Request completed successfully.</td>
</tr>
<tr>
<td>0</td>
<td>X'08'</td>
<td>Indicates a duplicate alternate key. For GET requests, this means that there are additional base records with the same alternate key. For PUT requests, indicates a duplicate entry for the alternate key was created.</td>
</tr>
<tr>
<td>4</td>
<td>X'00'</td>
<td>Indicates that the RPL was already active with another request.</td>
</tr>
<tr>
<td>08</td>
<td>X'04'</td>
<td>Logical Error Occurred (See Error Code for Reason)</td>
</tr>
<tr>
<td>08</td>
<td>X'08'</td>
<td>Logical End of File, there are no records with any higher key value than that of the last record returned. For a POINT, or START BROWSE, the key specified is higher than the highest key on the file.</td>
</tr>
<tr>
<td>08</td>
<td>X'0C'</td>
<td>Out of sequence.</td>
</tr>
<tr>
<td>08</td>
<td>X'10'</td>
<td>No record was found in the file with the specified key.</td>
</tr>
<tr>
<td>08</td>
<td>X'16'</td>
<td>A PUT to add a record was issued, however there was already a record on the file with that same key.</td>
</tr>
</tbody>
</table>

On a PUT during file load, the key of the record is lower than the previous record. Records must be loaded in ascending key sequence.

On a PUT in sequential mode (OPTCD=SEQ), the key of the record being added is lower than the key of the last record processed (either retrieved or added) by this RPL.

On a Skip Sequential request (POINT or GET), the key requested is lower than the key of the record previously retrieved.
# IAM I/O Request Error Codes

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Error Code (Decimal)</th>
<th>Error Code (Hex)</th>
<th>Description of Error Condition:</th>
</tr>
</thead>
<tbody>
<tr>
<td>08</td>
<td>20</td>
<td>X'14'</td>
<td>Record is under Exclusive Control: The same record has been requested for UPDATE by another RPL. The RPL message area, if provided, has the address of the RPL that holds the requested record for update. From IAM-RLS: The records is locked by another RPL for the same logical unit of work (batch job, TSO user, or CICS transaction).</td>
</tr>
<tr>
<td>08</td>
<td>21</td>
<td>X'15'</td>
<td>IAM-RLS detected that by waiting for this record lock, this request would have caused in a deadlock.</td>
</tr>
<tr>
<td>08</td>
<td>22</td>
<td>X'16'</td>
<td>IAM-RLS: Request timed out waiting for a record lock.</td>
</tr>
<tr>
<td>08</td>
<td>24</td>
<td>X'18'</td>
<td>IAM-RLS: Request was waiting for a record lock that was retained by another job due to an abend.</td>
</tr>
<tr>
<td>08</td>
<td>28</td>
<td>X'1C'</td>
<td>For Compatible format files, the Independent Overflow area is filled, the file needs to be reorganized. For Enhanced format files, there either is insufficient DASD space to expand the file, or the file has used up the maximum number of extents it is permitted.</td>
</tr>
<tr>
<td>08</td>
<td>32</td>
<td>X'20'</td>
<td>ESDS: RBA supplied does not specify the address of the beginning of a record. No record at the specified relative byte address.</td>
</tr>
<tr>
<td>08</td>
<td>40</td>
<td>X'28'</td>
<td>IAM was not able to obtain virtual storage to complete the request. This will normally only occur on a GET request, with OPTCD=LOC, where IAM could not obtain storage for an area to contain the requested record.</td>
</tr>
<tr>
<td>08</td>
<td>44</td>
<td>X'2C'</td>
<td>The area provided by the application program was not large enough to contain the requested record. The record size is in the RPL field RPLRLEN.</td>
</tr>
<tr>
<td>08</td>
<td>64</td>
<td>X'40'</td>
<td>For Enhanced Format files: IAM was unable to obtain virtual storage for an additional string (place holder). For Compatible Format files, an insufficient number was specified for STRNO, and IAM ran out of place holders.</td>
</tr>
<tr>
<td>08</td>
<td>68</td>
<td>X'44'</td>
<td>An UPDATE request was issued, i.e. PUT or ERASE, however the file was opened for INPUT processing only.</td>
</tr>
<tr>
<td>08</td>
<td>72</td>
<td>X'48'</td>
<td>Keyed access attempted on an ESDS type of file.</td>
</tr>
<tr>
<td>08</td>
<td>80</td>
<td>X'50'</td>
<td>Erase attempted on an ESDS type of file.</td>
</tr>
<tr>
<td>08</td>
<td>84</td>
<td>X'54'</td>
<td>PUT with locate mode (OPTCD=LOC) is not permitted.</td>
</tr>
<tr>
<td>08</td>
<td>88</td>
<td>X'58'</td>
<td>RPL is not positioned for the specified sequential request. A POINT is required, or a random get with positioning: OPTCD=(DIR,NSP).</td>
</tr>
<tr>
<td>08</td>
<td>92</td>
<td>X'5C'</td>
<td>A PUT or ERASE request was issued without a preceding GET for update.</td>
</tr>
<tr>
<td>08</td>
<td>96</td>
<td>X'60'</td>
<td>On an update PUT request, the key in the record does not match the key of the record read for update.</td>
</tr>
<tr>
<td>08</td>
<td>100</td>
<td>X'64'</td>
<td>ESDS file type: on an update PUT request, the user attempted to change the record length.</td>
</tr>
<tr>
<td>Return Code</td>
<td>Error Code (Decimal)</td>
<td>Error Code (Hex)</td>
<td>Description of Error Condition:</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------</td>
<td>------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>08</td>
<td>104</td>
<td>X'68'</td>
<td>Invalid RPL options specified. (OPTCD) KSDS: Relative Byte Address or Control Interval processing is not supported by IAM. (OPTCD=ADR or OPTCD=CNV) or a get previous request is issued in skip sequential mode (RPL OPTCD=(SKP,BWD)).</td>
</tr>
<tr>
<td>08</td>
<td>108</td>
<td>X'6C'</td>
<td>The record length either is less than the minimum record length, which is (key length + key offset), or exceeds the maximum defined record length. For compatible format IAM files with the FIXED length record attribute, the record length was not equal to the defined length.</td>
</tr>
<tr>
<td>08</td>
<td>112</td>
<td>X'70'</td>
<td>The key length in the RPL is greater than the defined length of the key. (RPL OPTCD=GEN type requests only.)</td>
</tr>
<tr>
<td>08</td>
<td>116</td>
<td>X'74'</td>
<td>The request type (i.e., a GET, POINT, or ERASE) is not valid during a file load.</td>
</tr>
<tr>
<td>08</td>
<td>144</td>
<td>X'90'</td>
<td>An invalid pointer was found in an alternate index. While the requested key was found in the alternate index, the associated base record was not found.</td>
</tr>
<tr>
<td>08</td>
<td>156</td>
<td>X'9C'</td>
<td>An internal IAM error was encountered while processing the request. IAM found a record with a length of zero, or reached the end of data within a block unexpectedly. This error could be due to a corrupted buffer in storage. Run an IAMRECVR DIAGNOSE to verify that the file is ok.</td>
</tr>
<tr>
<td>08</td>
<td>184</td>
<td>X'B8'</td>
<td>I/O request for IAM-RLS either failed or abended.</td>
</tr>
<tr>
<td>08</td>
<td>185</td>
<td>X'B9'</td>
<td>I/O request for IAM-RLS was rejected because either IAM-RLS was no longer active, or was being quiesced.</td>
</tr>
<tr>
<td>08</td>
<td>189</td>
<td>X'BD'</td>
<td>I/O request for IAM-RLS was rejected because the file had been closed under IAM-RLS by operator request.</td>
</tr>
<tr>
<td>08</td>
<td>192</td>
<td>X'C0'</td>
<td>The program issued a request specifying an invalid relative record number. This can primarily occur if a relative record number of 0.</td>
</tr>
<tr>
<td>08</td>
<td>200</td>
<td>X'C8'</td>
<td>A control interval request (OPTCD=CNV) is not permitted when reading an ESDS through an alternate index.</td>
</tr>
<tr>
<td>08</td>
<td>208</td>
<td>X'D0'</td>
<td>A CHECK or an ENDRREQ request was issued, however there was another task waiting on the RPL ECB.</td>
</tr>
<tr>
<td>08</td>
<td>225</td>
<td>X'E1'</td>
<td>Internal IAM error: The buffer pointer in the PLH is 0.</td>
</tr>
<tr>
<td>08</td>
<td>241</td>
<td>X'F1'</td>
<td>A invalid type of request was made, the contents of R0 contains a VSAM request type that IAM does not recognize or support.</td>
</tr>
<tr>
<td>08</td>
<td>242</td>
<td>X'F2'</td>
<td>The ECB address passed in the RPL is invalid.</td>
</tr>
<tr>
<td>08</td>
<td>244</td>
<td>X'F4'</td>
<td>During a file load, the Data Space used to temporarily hold the index was filled. Rerun job with the DATASPACE override, providing a larger value.</td>
</tr>
<tr>
<td>0C</td>
<td>4</td>
<td>X'04'</td>
<td>An error occurred attempting to READ a data block from DASD.</td>
</tr>
<tr>
<td>0C</td>
<td>16</td>
<td>X'10'</td>
<td>An error occurred when IAM was attempting to WRITE a data block to DASD.</td>
</tr>
</tbody>
</table>
### 80.22 IAM Open and Close Error Codes

In keeping with the VSAM application programming interface, most errors that occur during Open or Close will not cause an abend. Rather, a non-zero return code is passed in register 15, and an error code is set in the ACBERFLG field of the ACB. The error code can also be retrieved by the SHOWCB macro, requested for the failing ACB. Request FIELDS=ERROR in the SHOWCB macro to obtain the error code. Most of the IAM Open or Close failures will result in an IAMWxx error message being generated. Refer to the message for additional information.

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Error Code (Decimal)</th>
<th>Error Code (Hex)</th>
<th>Description of Error Condition:</th>
</tr>
</thead>
<tbody>
<tr>
<td>04</td>
<td>4</td>
<td>X'04'</td>
<td>A CLOSE was requested for an ACB that was already closed.</td>
</tr>
<tr>
<td>04</td>
<td>100</td>
<td>X'64'</td>
<td>This is a warning that on an open for a PATH or base cluster with an alternate index(es) in an upgrade set, one or more of the alternate index datasets were found to be empty.</td>
</tr>
<tr>
<td>04</td>
<td>136</td>
<td>X'88'</td>
<td>Insufficient storage in region to close the file. Most likely, there is insufficient below the line storage to obtain work area(s) for the file close. Perhaps by closing other files first, if possible, will correct this circumstance. It may be necessary to raise the value of the REGION parameter for this job step.</td>
</tr>
<tr>
<td>04</td>
<td>144</td>
<td>X'90'</td>
<td>The CLOSE for an IAM file being loaded or reorganized failed, due to insufficient DASD space to write out the file's index. Delete and redefine the file with more DASD space.</td>
</tr>
<tr>
<td>08</td>
<td>128</td>
<td>X'80'</td>
<td>DD Statement for the specified dataset is not in the JCL. This also will be issued when one of the macros issued by IAM to OPEN the dataset fails, such as OPEN, DEVTYPE, or TRKCALC.</td>
</tr>
<tr>
<td>08</td>
<td>136</td>
<td>X'88'</td>
<td>Insufficient storage in region to open the file. Most likely, there is insufficient extended private storage, however there could also be insufficient below the line storage. Check message IEF374I to determine how much storage was used. Most likely increasing the REGION parameter will solve the problem.</td>
</tr>
<tr>
<td>08</td>
<td>152</td>
<td>X'98'</td>
<td>RACF, or other system security software, indicated that the user was not allowed the type of access to the file that was being requested by the ACB on the OPEN macro.</td>
</tr>
<tr>
<td>08</td>
<td>160</td>
<td>X'A0'</td>
<td>The application attempted to open an IAM file for input processing, however either the file had never been loaded, or an attempted file load or reorganization failed. Or, the application attempted to open an unloaded file with a STRNO value not equal to 1.</td>
</tr>
<tr>
<td>Return Code</td>
<td>Error Code (Decimal)</td>
<td>Error Code (Hex)</td>
<td>Description of Error Condition:</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------</td>
<td>-----------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>08</td>
<td>167</td>
<td>X'AF'</td>
<td>IAM-RLS has encountered an abend attempting to open or close this dataset.</td>
</tr>
<tr>
<td>08</td>
<td>168</td>
<td>X'BF'</td>
<td>The specified file was already being processed by another job or user, and could not be opened for the processing that the application requested. Check for an IAMW04 or IAMW30, which will, when possible, include the name of the job that had the file opened. Rerun the failing job after the identified job terminates.</td>
</tr>
<tr>
<td>08</td>
<td>182</td>
<td>X'BD'</td>
<td>The Open was failed because the IAM-RLS was not active.</td>
</tr>
<tr>
<td>08</td>
<td>184</td>
<td>X'BD'</td>
<td>An I/O error occurred while trying to open the indicated file. This error code should be accompanied by either an IAMW37 or an IAMW07 error message, indicating the nature of the I/O error.</td>
</tr>
<tr>
<td>08</td>
<td>188</td>
<td>X'BD'</td>
<td>Open processing encountered a problem while attempting to Open the file. If the file was being loaded, an invalid or unsupported record length was specified for the file. This error may also be due a problem with the extended data areas of the file. There should be an IAMWxx message indicating the reason for the error.</td>
</tr>
<tr>
<td>08</td>
<td>192</td>
<td>X'CF'</td>
<td>Open processing failed for a file load, due to an invalid file attribute being specified. This should be accompanied by an IAMW20 error message, indicating the improperly specified attribute.</td>
</tr>
<tr>
<td>08</td>
<td>196</td>
<td>X'CF'</td>
<td>Open processing failed on a PATH because the alternate index object of that PATH was empty.</td>
</tr>
<tr>
<td>08</td>
<td>209</td>
<td>X'CF'</td>
<td>The open for this file failed under IAM-RLS. Review the IAM-RLS log for additional information.</td>
</tr>
<tr>
<td>08</td>
<td>232</td>
<td>X'CF'</td>
<td>An attempt was made to Open a loaded IAM file for reloading with ACB MACRF=RST, however the file was defined with the NOREUSE attribute, and the IAM Global Option was set to ENABLE=NOREUSE. The file must be deleted and redefined to be reloaded.</td>
</tr>
<tr>
<td>08</td>
<td>240</td>
<td>X'CF'</td>
<td>The file was already open by another ACB (or DD) in this address space, and the current attempt to OPEN the file failed because the time stamp for file load was different than the time stamp in the previously opened ACB. Close all open ACB for this dataset in this address space, and then they can be reopened.</td>
</tr>
<tr>
<td>08</td>
<td>241</td>
<td>X'CF'</td>
<td>The IAM VSAM Interface was not able to find a proper version of the IAM processing module required to OPEN this file.</td>
</tr>
<tr>
<td>08</td>
<td>242</td>
<td>X'CF'</td>
<td>The file was being opened for OUTPUT with an expired trial version of IAM.</td>
</tr>
<tr>
<td>08</td>
<td>248</td>
<td>X'CF'</td>
<td>Internal IAM error, the Open request failed.</td>
</tr>
<tr>
<td>08</td>
<td>254</td>
<td>X'CF'</td>
<td>The IAM VSAM Interface either was unable to successfully LOAD the required IAM processing module, or a version was loaded that does not match the version of the IAM VSAM Interface (VIF) that is active.</td>
</tr>
</tbody>
</table>
### 80.23 COBOL File Status Codes with IAM

While the above error codes are available to COBOL programs, more frequently these programs use the COBOL File Status Codes for interpreting error situations. Below are the more frequently encountered File Status Codes which can occur while processing an IAM file.

<table>
<thead>
<tr>
<th>File Status</th>
<th>Reason for File Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Successful completion of request.</td>
</tr>
<tr>
<td>02</td>
<td>Successful completion of request. There are additional base records with the same alternate key.</td>
</tr>
<tr>
<td>04</td>
<td>Successful completion of request, however the record size does not match the fixed length of the record as defined within the COBOL program.</td>
</tr>
<tr>
<td>10</td>
<td>End of file.</td>
</tr>
<tr>
<td>14</td>
<td>Relative Record Number given exceeds maximum for this file.</td>
</tr>
<tr>
<td>21</td>
<td>On sequential WRITES, the key is lower than the previously written key. Or, on REWRITE of an existing record, the application program changed the key of the record.</td>
</tr>
<tr>
<td>22</td>
<td>A WRITE of a new record was attempted for a key that matches a record currently in the file.</td>
</tr>
<tr>
<td>23</td>
<td>The requested record (key) was not found in the file.</td>
</tr>
<tr>
<td>24</td>
<td>Additional DASD space was required for this request, but it could not be obtained. For Compatible format files, this indicates that the Independent Overflow area has been filled. For Enhanced format files, either there was insufficient DASD space to expand the file, or the file had reached the maximum extents allowed, which is 16 extents per DASD volume. Generally, this error requires that the file be reorganized, possibly requiring a DELETE and DEFINE to allocate more DASD space. For Enhanced Format files, if the cause is insufficient DASD space, if some existing datasets can be deleted from the DASD volume, it might be possible to subsequently retry the request.</td>
</tr>
<tr>
<td>34</td>
<td>The file is full. See description for file status code 24 above.</td>
</tr>
<tr>
<td>35</td>
<td>Improper OPEN of an empty (unloaded) file, such as open for input only.</td>
</tr>
<tr>
<td>37</td>
<td>Invalid mode for OPEN</td>
</tr>
<tr>
<td>39</td>
<td>This file status generally implies a mismatch between the defined IAM file attributes and the record layout in the COBOL program. For example, possible causes are the key length or offset from the file definition do not match the COBOL record layout. Also, the defined maximum record length for the file is shorter than the maximum possible length for variable length records. COBOL requires that the defined record length be at least as long as the maximum theoretical record size, from the layout. Refer to the IAM PSEUDOLRECL feature for circumventing this restriction.</td>
</tr>
<tr>
<td>41</td>
<td>An OPEN was requested for a file that had already been opened.</td>
</tr>
<tr>
<td>42</td>
<td>A CLOSE was issued for a file that was already closed.</td>
</tr>
<tr>
<td>43</td>
<td>A file update request (i.e., REWRITE or DELETE) was issued without a prior READ for UPDATE.</td>
</tr>
<tr>
<td>44</td>
<td>An incorrect record length was specified on the WRITE. Either the length was shorter than the minimum, which is key length plus key offset, or the record was longer than the defined maximum record size for the file.</td>
</tr>
<tr>
<td>File Status</td>
<td>Reason for File Status Code</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>46</td>
<td>A READ failed because the application had not successfully established a position in the file, or a READ was attempted after the end of file had been reached.</td>
</tr>
<tr>
<td>47</td>
<td>A READ request was issued for a file that was not opened for input or I/O.</td>
</tr>
<tr>
<td>48</td>
<td>A WRITE request was issued for a file not opened for output or I/O (update).</td>
</tr>
<tr>
<td>49</td>
<td>A DELETE or REWRITE request was issued for a file that was not opened for I/O (update).</td>
</tr>
<tr>
<td>90</td>
<td>Logical error. Possible IAM logical error code x’E1’.</td>
</tr>
<tr>
<td>92</td>
<td>Logic error. The file attributes from the COBOL program, such as key length, relative key position, or record length do not match the IAM file being accessed. (Note that the defined record length for the file must be at least as large as the largest possible record length.)</td>
</tr>
<tr>
<td>93</td>
<td>Record lock contention or exclusive control errors. Also could be caused by insufficient virtual storage to process the request. Most likely, the REGION parameter has to be increased for the job. Check the IAM output, IAMPRINT, from a Listcat of the file to determine how much storage will be required to open this file.</td>
</tr>
<tr>
<td>94</td>
<td>Not positioned for a sequential READ request.</td>
</tr>
<tr>
<td>95</td>
<td>Invalid or incomplete file information.</td>
</tr>
<tr>
<td>96</td>
<td>The DD statement is missing for this file.</td>
</tr>
</tbody>
</table>
90 IAM INSTALLATION INSTRUCTIONS

90.01 IAM Installation Introduction

IAM should be installed using the interactive installation procedure documented here in Section 90. This requires TSO with ISPF.

Innovation strongly recommends the use of this interactive installation procedure. It is easier to use and less prone to errors than any manual procedures.

The installation of IAM can be done in three different ways:

**Electronic Installation** – The install package is downloaded from the Innovation FTP site. If you are installing the product using this method, follow the instructions in Section 90.02.

**CD Installation** – The install package is downloaded from a CD. If you are installing the product using this method, follow the instructions in Section 90.03.

**Load Library**

Several load modules that were included with the distribution with prior releases have been removed for the IAM Version 9.2 distribution. These modules were primarily for the old IAM - ISAM interface that to our knowledge not been used or supported for several years. Please contact Innovation technical support if there are any questions regarding the load modules that are no longer being shipped.

**CAUTIONS**

IAM Version 9.2 requires an IBM z/OS or higher level operating system and requires a processor that supports the z/Architecture Principle of Operations -04 level (fifth edition) and above, which are for the z/890 and z990 and above. IAM support on systems that do not meet those requirements and are using z/Architecture machine can use up through Version 9.1, and for systems without z/Architecture support is provided by IAM Version 8.1.

Do not copy the IAM Global Options from a prior version. Please follow the procedure in Section 90.10 to migrate from the IAM version you are currently running up to IAM Version 9.2.

Please note that the IAM PRO file format in IAM Version 9.2 is downward compatible with Version 9.1 or 9.0, however they are not downward compatible to IAM Version 8.1 or prior releases. Any PRO files created in IAM Version 9.0 or higher will have to be converted to an Enhanced format IAM file with IAM Version 9.0 or Version 9.1 prior to using it with IAM V8.1 or prior version.

Please note that IAM Version 9.2 support of IAM files on EAV volumes are not downward compatible with IAM Version 8.1 or prior. Also note that IAM files on EAV volumes have minimum operating system levels of z/OS 1.11 for SMS Extended Format IAM files, and z/OS 1.12 for basic sequential and large sequential IAM files.
90.02 Electronic Installation

**STEP 1.1**

With the electronic installation, instructions will be sent via a set of two e-mails for installing the libraries. The e-mail installation instructions supersede these printed instructions and should be used if any differences are noted.

The order acknowledgment e-mail (first of two e-mails) will look similar to:

```
Subject: Innovation Distribution Server Order Acknowledgment 012345
From: <ENSMTP@CPUA.IDPNJ.COM>
Date: mm/dd/20yy hh:mm xM
To: SYSPROG@ABC.Company

You are receiving the first of a two message set acknowledging your Innovation software product order. Retain this e-mail until you receive a status (second) message containing instructions for performing the product installation. The status message will be sent when your order is ready to be downloaded from the Innovation FTP site.

NOTE:
1. You must perform the install of this distribution file before nnnDAY month day year.
The install program IDPREC in your distribution file will expire on this day and the file will be removed from our FTP site.
2. Hardware and Software requirements:
   This software must be installed on a z/Architecture processor and execute on a z/OS Operating System.
The load library must be a PDSE.

This order is for IAM Product Group including the following options:
IAM AIX SPX

Customer Order Number:
02501001234531620
Company name:
ABC COMPANY
Customer contact:
SYS PROG
Product version:
9.2/00
IDPREC Password:
12345
```

*Figure 90-1: Example of first of two e-mails for Electronic Installation*

This first e-mail contains the products and versions that are included in the install package, the expiration date (if the product has an expiration date), and a unique IDPREC password that is required for the install process.
STEP 1.2 The order completion e-mail (second of 2 e-mails) will look similar to:

Subject: Innovation Distribution Server Order Completion 012345
From: <ENSMTP@CPUA.IDPNJ.COM>
Date:mm/dd/20yy hh:mm xM
To: SYSPROG@ABC.COMPANY

You are receiving the second of a two message set acknowledging your Innovation software product order is ready to be downloaded from the Innovation FTP site. Verify that the Customer Order Number is the same on both e?mail messages before performing the retrieval procedure below.

This order is for IAM Product Group including the following options:
IAM AIX SPX

Customer Order Number:
02501001234531620
Customer number:
012345
Company name:
ABC COMPANY
Customer contact:
SYS PROG
Product version:
9.2/00
IDPREC Password:
12345

The instructions below outline the procedure you will use to retrieve the product file from the Innovation FTP site and then install our software.

The most convenient method is to click on the HTML link below or paste this URL into the Address Line of your browser. This link will start a Java applet(J2SE 1.5 or higher required) that will download a product distribution file from the Innovation FTP site to your PC using the Windows FTP EXE client, then connect to your z/OS FTP server to transfer this file to z/OS. This FTP transfer is done within the user’s local network, data is never transmitted across the Internet. You will need to know your z/OS host connection information (Host IP or DNS name, TSO Userid and password). The first page to display will be an introduction page. To continue with the product installation click on “Enter” and when the next page displays for more information click on the button labeled “Getting Started”. ......

Figure 90-2: Second of two e-mails for Electronic Installation

This second e-mail contains the links to a unique ftp site containing the product installation package. The transfer of the installation package can be done with a Java applet or via manual ftp download if you encounter problems running the Java applet at your site. Instructions are included in this e-mail for both methods.
STEP 1.3 The Java applet installation procedure is the easiest and will download a product distribution file from the Innovation FTP site to your PC using the Windows FTP.EXE client. It will then connect to your MVS FTP server to transfer this file to MVS. This process is done within the user’s local network; data is never transmitted across the internet.

The first screen you will see when executing the Java applet is shown below:

![Figure 90-3: Initial screen when starting the IAM Product Installation.](image-url)
Click the ENTER button. Depending on your settings, you may see the following screen first:

![Image of prompting screen to run JAVA application]

Figure 90-4: Possible prompting screen to run the JAVA application.
After you click on run, then the following screen will appear.

![IAM Product Installation Screen](image)

Figure 90-5: IAM Product Installation Screen

Clicking on the **Getting Started...** button will display a window with further instructions, such as is shown below:
The instructions that follow were current at the time of publication. Always review the instructions provided as they will contain the current instructions which supersede the instructions below.
After filling in the required information, and making any desired changes, the screen may look something like this:

![IAM Product Install Window](image)

*Figure 90-7: Example of a completed IAM Product Install window.*
Now you can click the **START TRANSFER** button. The following window will be displayed:

![Confirm Transfer](image)

*Figure 90-8: Confirmation of target dataset file name.*

If the dataset name is acceptable, click the **YES** button to continue. If not, click the **NO** button and you will go back to the prior window and be able to change the information. I went back to change the dataset name to 'RAM.IAM.BIN' and then replied clicked YES.

Upon successful file transfer to your MVS host, the applet will display a confirmation message box labeled “Transfer to MVS Host Successful”.

![Transfer To MVS Host Successful](image)

*Figure 90-9: Notification of successful FTP transfer to host system.*

After clicking “OK”, another message box containing the remaining MVS installation instructions will be displayed.
Click on the Print button to print out the full instructions. The instructions in the manual were current at the time of publication, however the provided instructions will supersede the instructions provided here.

If the JAVA applet process has been successful you can skip Step 1.4, and continue with Step 1.5.

**STEP 1.4** If you have encountered problems with the JAVA applet, or are not able to use that process, obtain the manual FTP instructions as directed from the second e-mail. Those instructions will always reflect the information current at the time you execute this process. Once you have successfully transferred the IAM.BIN file to your z/OS main frame, you can continue with Step 1.5.
STEP 1.5  Follow the “Remaining Product Installation Instructions” that are presented at the end of the file transfer. Expanding the product distribution file on your MVS host is a two-step process. If you are using ISPF, issue the following TSO commands from ISPF Option 6 (TSO Commands). You can also exit ISPF and issue them from the TSO “READY” prompt. This TSO session must have access to an ISPF environment because the MVS install process expects to employ ISPF panels to complete the installation. The steps described below assume TSO PROFILE NOPREFIX is NOT used.

1. RECEIVE INDATASET(IAM.BIN)

   Where the INDATASET(...) value is the name that you created during the file transfer to MVS.

   Figure 90-11:  ISPF Option 6 with RECEIVE command entered.
This creates a partitioned dataset containing two load modules. After the RECEIVE is completed, the name of the PDS will be ‘userid.IDPREC.LOAD’. If you want a different name, then at the prompt:

Enter:  **DSN(desired.name)**

Or take the default by pressing ENTER. The following screen will appear with the name of the PDS that was loaded.
2. CALL 'userid.IDPREC.LOAD,IDPREC' 
   or, if you gave the PDS a different name:

   CALL 'dsname(IDPREC)'

   ```
   Menu List Mode Functions Utilities Help
   ISPF Command Shell
   ISPF Command =>
   Enter TSO or Workstation commands below:
   => call idprec.load(idprec)
   \n   Place cursor on choice and press enter to Delete command
   => RECEIVE INDATASET(IAM.BIN)
   =>
   =>
   =>
   =>
   =>
   =>
   =>
   =>
   ```
A welcome message is displayed. Enter appropriate responses to name and sysout class prompts. You will also be prompted to enter the 16 digit IDPREC PASSWORD supplied in the Order Acknowledgement e-mail. Screen below shows this sequence with taking defaults at all of the prompts.

Figure 90-12: Screen showing prompt for the IDPREC password to continue with install.

After entering the valid 16 digit password, the IDP Product Install SCREEN1 will appear, as shown on the next page. This ISPF panel will enable you to revise the names of the datasets that will be allocated as part of the product installation. Any entry that indicates NOT.SHIPPED are not a part of the IAM product distribution.
STEP 1.6 SCREEN 1 – Dataset Name Selection

Figure 90-13: IAM Installation Screen 1 (top half)

Figure 90-14: IAM Installation Screen 1 (bottom half)
The screen(s) on the prior page allows you to specify the dataset names which will be used for the datasets you are loading from the install package. These may be existing datasets to be updated, or they may be new datasets that will be allocated and cataloged (new datasets are recommended).

The load library must be an APF authorized library. If necessary, you can authorize it after the install using the SETPROG console command (see the IBM “System Commands” manual for details).

The names shown above are the default names provided by the IDPREC Install program (when the TSO userid of RAM was used). You may change these names in one of two ways:

To change the High or 2nd Level qualifiers for all datasets, change the values in the corresponding field for the Output DSN section.

To change an individual Low Level qualifier, change the desired Low Level Qualifier for the dataset(s) that you want to change.

The names of the resulting dataset names that will be used are displayed in the bottom area of the panel.

You can also specify the SMS attributes (if they are to be SMS-managed) or VOLUME/UNIT information on this panel to allocate these datasets when creating them as NEW.
STEP 1.7  SCREEN2 – IDP PRODUCT INSTALL

This screen documents the process that will be used to create the product libraries. When you press ENTER to continue, it will also show you the status of each of the datasets being created. Once the product libraries have been created, a completion message will be displayed.

Figure 90-15: IAM Installation Screen 2

Figure 90-16: IAM Installation Screen 2 - Completion Acknowledged
STEP 1.8  SCREEN 3 – Installation Datasets Have Been Received

Figure 90-17: IAM Installation Screen 3

This screen documents that all the product libraries have been created and received and you can now go to the Primary Options Menu to continue the installation process and customize the product.

Proceed to Section 90.10 to complete the installation.
90.03 CD Installation

STEP 2.1 The CD install method contains a letter that documents the install process. (A copy of this letter is in electronic form in the Readme.rtf file in the zOS directory found in the root directory of the CD). You should follow these instructions provided to install the product using the CD install method. You must transfer the IAM.BIN file from the zOS directory found in the root directory of the CD to your MVS host. The required file attributes for the target MVS dataset will be listed in the instructions. The exact directions for the FTP transfer are also documented in the instructions.

STEP 2.2 Once the IAM.BIN file is on the MVS host system, you will then expand the product distribution file. From this point forward follow the instructions starting at STEP 1.5 in section 90.02 Electronic Installation. The rest of the CD installation matches the Electronic Installation starting from STEP 1.5.
Setting the IAM Global Options

Overview

The next step is to review and make whatever changes are necessary to the IAM Global Options. Many of the various default values for IAM can be changed in the IAM Global Options Table. For example, defaults for buffers, Data Compression, SMF recording, and many others can be changed using this capability. The defaults are changed by executing the IAMZAPOP utility program. The use of this program and all of the default values and options are fully described in Section 91 of the IAM User Manual. This section will discuss what is new and changed in this version, how to use the IAM Global Options change capability along with some examples, plus make some recommendations for changing some of the defaults.

New in IAM V9.2

The following are the new and changed Global Options for Version 9.2:

- **CRBUFOPT=64BIT** The 64BIT is a new operand for the CRBUFOPT Global Option that specifies that the IAM file load process will acquire storage for two cylinders of buffers in above the bar (64-bit addressable virtual storage). The default value for CRBUFOPT as IAM is shipped remains as MCYL, which is two cylinders of buffers in 31-bit addressable storage. Note that the use of 64BIT will cause an increase in CPU time over the MCYL option that is primarily from the overhead of fixing the pages of 64-bit virtual storage.

- **DATASPACE=2048** The default for this Global Option has been changed from 1024 to 2048, the maximum value permitted. This change was made because some users have encountered problems loading very large files that ended up failing due to insufficient data space storage with the prior default value.

- **ENABLE or DISABLE= BUF64** This option when enabled specifies that IAM will attempt to acquire storage for buffers during normal access of IAM Enhanced Format files in 64-bit virtual storage. The default is that this is disabled, that IAM will use 31-bit virtual storage. Note that use of BUF64 will increase CPU time due primarily to overhead of fixing the pages of 64-bit virtual storage.

- **ENABLE or DISABLE= HPF** This option when enabled specifies that IAM will automatically use z/HPF channel programs for volumes that the IOSZHPF macro indicates support such channel programs. The use of z/HPF channel programs when available enhances channel performance and may result in reduced time to perform I/O operations. The default is that HPF is enabled. The option to disable use of z/HPF is provided in case that usage is causing performance or functional problems.

- **IAMW22LIM=nnnn** Specifies up to a 4-digit number representing megabytes of storage used for the overflow index of an IAM data set. When more than the specified amount of storage is in use for any particular data set an IAMW22 message will be displayed indicating that a reorganization of the file may be necessary. The default value is 64. In prior releases the message appeared when the overflow index exceeded 16 megabytes.

New in IAM V9.1

The new and changed Global Options for Version 9.1 of IAM are:

- **CICSTXID=cccc** - This option sets the name of transaction IAM is to use for the IAM provided transaction to facilitate that IAM/RLS and IAM/PLEX functions for automatic disconnect and reconnect should those address spaces become unavailable. The default value is $IAM.

- **ENABLE=SAFRC4 or DISABLE=SAFRC4** - This option when enabled will result in an IAMW28 warning message being issued if the security call issued by
IAM when opening a dataset indicated that no security decision was made, that is it was completed with a return code of 4. This option is provided to help ensure that your security software environment is functioning properly with IAM.

**New in IAM V9.0**

The following options are those that were new and changed for Version 9.0, is provided for those who are upgrading from IAM Version 8.1:

- **ENABLE=XESDS or DISABLE=XESDS** - This option will enable or disable 8-byte RBA's as the default for ESDS files under IAM. If disabled then 4 byte RBA's will be used unless overridden with the XESDS CREATE override.

- **ENABLE=ESDSLOCK or DISABLE=ESDSLOCK** - A new option to control the CI lock obtained on the high used RBA in an ESDS file while adding records. If any applications are adding records to an ESDS file and the applications are dependent on the order in which these adds are done, this option should be enabled. An exclusive control reason code will be returned to the application if the CI is locked and it can reissue the request. If the order of adds is not important then this option can be disabled to eliminate the exclusive control waits on the last CI.

- **BELOWPOOL=YES** - The default for this option has been changed to YES in V9.1. Many of our users have been using this option to reduce the below the line storage used by IAM file in CICS regions.

- **DATASPACE=1024** - The default for this option has been changed from 256 to 1024 to allow for the maximum DATASPACE size of 2gig used for the INDEXSPACE.

- **INDEXSPACE=64BIT** - The default for this option has been changed from CICS to 64BIT which will cause above the bar 64BIT storage to be used for the INDEXSPACE if above the bar storage can be allocated. Depending on your installation you may need to specify a MEMLIMIT keyword on the EXEC card in job steps where you want to use the 64BIT INDEXSPACE. If above the bar storage cannot be obtained then a DATASPACE will be obtained for the INDEXSPACE instead. Changing this option to ALL or CICS will cause the INDEXSPACE to be used only and will bypass any 64BIT processing.

- **LONGKEYS=33** - The default for this option has been changed from 128 to 33 as we have found this to be a better fit for most of our users.

**Users of Prior Versions of IAM**

The first step in determining what, if any, Global Options should be changed is to find out what Global Options were changed previously. Users of prior versions can determine what Global Options they have changed in their current release through the use of the IAMZAPOP AUDIT function. This is a critical part of installing a new release or level of IAM to insure that existing applications will continue to run as they have with the prior levels. Below is an example of running the AUDIT function.

**WARNING: DO NOT COPY THE IAM GLOBAL OPTIONS FROM A PRIOR VERSION BECAUSE THE OPTION TABLE HAS CHANGED. PLEASE USE THE PROCEDURE DESCRIBED HERE.**
Example of IAMZAPOP AUDIT Command

```bash
//AUDIT EXEC PGM=IAMZAPOP
//STEPLIB DD DISP=SHR,DSN=current.iam.loadlib
//SYSLIB DD DISP=SHR,DSN=current.iam.loadlib
//SYSPRINT DD SYSOUT=*AUDIT
//SYGIN DD *
/*
```

Figure 90-18: JCL to run AUDIT Function of IAMZAPOP (EX9004A)

AUDIT Command Output

An example of the output from the AUDIT command follows. Note that it prints the name of the load library. The changed Global Option(s) keywords are displayed, followed by the current value, and the default value. From this example, we see that the following Global Options were changed:

- RECTYPE=201
- SMF=YES

These options are important to maintain across releases, as they provide for IAM SMF data to be collected, and enable customers to obtain reports on IAM activity using the IAMSFM or IAMSMFVS programs, or their own programs.

```plaintext
IAM400 IAM GLOBAL OPTION(S) CHANGE -- IAMZAPOP VER 9.2/01P -- INNOVATION DATA PROCESSING DATE - 2010.178
IAM491 AUDIT OPTIONS FUNCTION STARTED - 08.12.31
IAM541 THE FOLLOWING OPTIONS HAVE BEEN CHANGED IN -- IAMX.TESTLIB
  RECTYPE           SMF RECORD TYPE ---------------------------- 201 DEFAULT----------000
  SMF               SMF RECORDING ENABLED -----------------------YES DEFAULT----------NO
IAMS47 AUDIT REQUEST COMPLETE FOR --- IAMX.TESTLIB
IAMS492 AUDIT OPTIONS FUNCTION ENDED - 08.12.31 - CONDITION CODE 000
IAMS499 IAMZAPOP(9.2/01P ) PROCESSING COMPLETED
```

Figure 90-19: Example of Output from AUDIT Command

Recommended Global Option Changes

The next step, after determining what Global Options had been changed for the prior version, is to review the various default values and determine which, if any, would provide a benefit to your installation. Section 91 of the IAM User Manual describes the various options available. Innovation recommends that you consider changing the following Global Option values from their default settings.

- Set ESDSINTEGRATED=5 or higher to allow for updates to ESDS files to remain in their original block. While VSAM (and IAM) do not allow a record length to change on update processing for ESDS files, the record lengths can change due to data compression. It is better for overall performance to minimize the use of overflow, as much as is possible which is accomplished by defaulting to providing some integrated overflow for updated ESDS files.
- Setting the IAM Global Options -

- Set **SMF=YES** and **RECTYPE=201** to enable the IAM SMF records. This will provide a way, through SMF reports, to monitor and track IAM dataset usage. With this option enabled, the IAMINFO reports will be available from the IAM SMF records. This may prevent rerunning jobs just to obtain an IAMINFO report, should one be necessary. Valid record types are from 128 to 255. Make sure to select a record type that is not being produced by other software packages.

There are also a few options that you may have to set for compatibility with other software products. In particular, the following options may have to be set as indicated:

- Set **DYNCAT=YES** if you are using the BMC MAINVIEW product.
- Set **ENABLE=BIM** if you have any of the VSAM products from B. I. Moyle.
- Set **ENABLE=VAM** if you have CA-ALLOCATE.
- Set **ENABLE=SAFRC4** if you are using a product other than RACF for security, such as ACF2 or Top Secret to make sure that security decisions are being when opening data sets.
- Leave **DSORG=PS** set, particularly if you use DFSMSdss or DFSMShsm to backup your datasets. If IAM files have a DSORG of DA set, these products may not be able to restore IAM datasets, particularly if they are multi-volume.

Other Global Option Considerations

There are a few other Global Options that should also be considered, which may need to be set depending on your installation requirements. These include the following:

- While rather rare, there are some applications that require the setting of the Global Option **ENABLE=NOREUSE** for proper function. Such applications require an open error to occur when opening a non-reusable VSAM dataset to prevent overlaying a previously loaded file. By setting this Global Option, IAM will follow VSAM rules for reloading datasets without a Delete and Define as based on the Define parameters.
- If you already have CICS regions of 512 megabytes or higher, then consider raising the default value for **MAXREGION** to allow IAM to increase the region of those CICS systems if necessary. With a default value of 512 megabytes, IAM will not be able to increase the region for CICS systems that already have the large region value.
- If X37 abends have been a problem, consider revising the **MAXSECONDARY** default values. This facility acts as a multiplier that IAM will use when requesting another extent. Because IAM files that are not DFSMS Extended Format are limited to 16 extents per volume, use of this facility will increase the amount of DASD space requested without having to change the IDCAMS DEFINE. This option is not used for IAM files that are in the DFSMS Extended Format.
- Consider how you prefer allocations to work for multivolume datasets. Depending on this option, IAM will request either the primary or secondary space when it appears that a volume switch will occur. The keyword is **MULTIVOLUME=[ PRIMARY | SECONDARY]**. Because IAM does not have control over whether or not a volume switch occurs, this may not always work as desired, particularly when the primary space value is less than the secondary. This option is not used for IAM files that are in the DFSMS Extended Format.
- If you have a DFSMS STORCLAS that causes datasets to not be SMS managed, such as NONSMS, then set the IAM Global Option **STORCLAS** to that value.
Once you have decided on the Global Options you want to change, use the program IAMZAPOP to change them. Global Options can also be changed subsequent to the product installation as may be needed. The following JCL and control card example demonstrates how to set the recommended Global Option values. Refer to Section 91 for complete information on setting and changing the IAM Global Options Table.

```
//SETGLOPT EXEC PGM=IAMZAPOP
//STEPLIB DD DISP=SHR,DSN=new.iam.loadlib
//SYSLIB DD DISP=SHR,DSN=new.iam.loadlib
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
   ZAP ESDSINTEGRATED=5,SMF=YES,RECTYPE=201
   PRINT
/*
```

*Figure 90-20: Example of setting Recommended Global Option Values (EX9004B)*

The above example also includes a PRINT command. This will print out the Global Option settings, and can be done whenever a list is necessary. It is a good idea to review this listing, to make sure that the Global Options are set as you expect them to be. The PRINT command does not need to be preceded by the ZAP command, so a listing of your Global Options can be obtained whenever necessary. This listing of your Global Options can also be obtained through the IAM ISPF panels.
90.11 Installing the IAM ISPF Dialog

While completely optional, customers are encouraged to install the IAM ISPF Dialog. The dialog includes some utility functions that may be quite useful for some quick problem diagnosis. They also provide panels to display information about IAM file activity in CICS regions and within IAMRLS. The IAM ISPF Dialog Panels and messages are contained in 2 libraries on the installation tape. The libraries were loaded previously.

If you have installed the IAM panels and messages in libraries that are not specified in the ISPPLIB and ISPMLIB DD statements in the appropriate TSO logon procs, then you must add the dataset name specified for the panel library to the ISPPLIB concatenation, and add the dataset name specified for the message library to the ISPMLIB concatenation.

You must add an option to the ISPF/PDF primary option menu (ISR@PRIM), or an option menu of your choice, to invoke the IAM ISPF DIALOG program. Add a line that describes the IAM ISPF DIALOG to the PANEL BODY.

Example:

`% I +IAM – IAM Dataset Utilities`

Add a line that selects the IAMISPF program to the lines that translate the user entered options into the appropriate panel or program name in the PROC section of the panel definition.

```csh
&ZSEL = TRANS( TRUNC (&ZCMD,'.')
  0,'PANEL(ISPOPTA)'
  1,'PGM(ISRBRO)'
  .
  other panel options
  .
  I,'PGM(IAMISPF)' ← Add
  .
  other panel options
  ;',', '
  X,'EXIT'
  *,'?' )
```

*Figure 90-21: Example of modifying an ISPF Menu Panel to include IAMISPF*
Add the IAM Load Library to the TSO Logon Procs

If IAM has not been installed in a linklist library, you must make the IAMISPF load module, and other required IAM system modules, available under ISPF by adding the IAM load library to the STEPLIB DD statement in the appropriate logon procs. Please note that the use of the ISPLLIB DD for this purpose is strongly discouraged. If any of the IAM utility programs are executed from within IAMISPF via option U, they will NOT be loaded from the ISPLLIB DD. Additionally, some of the IAM utility functions require APF authorization which will require that all libraries specified in the STEPLIB DD be authorized.

APF Authorization for the IAM ISPF Dialog

The IAMRECVR DIAGNOSE and PRINT functions available under option U.R utilize system facilities that require APF (Authorized Program Facility) authorization. The IAMXMONA program, which displays information about IAM files in CICS regions also requires APF authorization. If any of these functions are used from within the IAM ISPF DIALOG, you must provide authorization for those utility programs as discussed below. If you do not wish to execute these functions under the IAM ISPF DIALOG, you will not have to provide APF authorization as described below.

IEAAPFx

The IAM load modules must be available to the IAM ISPF DIALOG via an APF authorized library.

The IAM load module library must be either:

- A linklist library that is APF authorized, or
- Allocated to the STEPLIB DD in the LOGON PROC.

Additionally, the IAM load module library and any other libraries that may be concatenated under the STEPLIB DD MUST be listed in the IEAAPFx member of SYS1.PARMLIB even if they are also in the LNKLSTFx member. Please note that you can NOT use the ISPLLIB DD for this purpose.

TSO Authorized Program Table

The two modules IAMRECVR and IAMXMONA must be set as authorized programs that are allowed to run with APF authorization under TSO. This can be done by either adding them to the IKJTSOxx member in the z/OS parmlib, or by changing the TSO authorized program lists. Changing the TSO authorized program lists are documented in the IBM manual "z/OS TSO/E Customization". The CSECTS IKJEFTAP in module IKJTABLES will need to have the two IAM module names indicated above added to those tables.

Updating IKJTSO00

If your installation uses the IKJTSO00 member, then add IAMRECVR and IAMXMONA to the AUTHTSF NAMES list. An IPL or PARMLIB command is required in order for the changes to IKJTSOxx to take effect.
90.12 Installing the IAM CICS Monitor

IAMXMON is a program that executes as a CICS transaction provided as part of the IAM product. This transaction displays activity and statistics for opened, Enhanced Format IAM datasets within the CICS region, similar to the IAMINFO reports that are produced when an IAM file is closed. For additional information on using IAMXMON refer to Section 10.75 of the IAM Manual.

After completing the following tasks, restart CICS, or INSTALL the transaction, program and map entries using CEDA

IAMXMON consists of a single program and a single mapset:

Program: IAMXMON language (Assembler) EXECKEY(CICS)

Mapset: IAMXMAP

These modules must be copied from the IAM LOADLIB into a LOADLIB that is part of the DFHRPL concatenation in order to work properly. IAMXMON is a CICS application, not part of the IAM access method code. It must be defined with EXECKEY(CICS). A transaction ID must be assigned to invoke the IAMXMON program, such as IAMX or IMON and be defined with TASKDATAKEY(CICS).

The details of installing IAMXMON are provided below:

JCL to Copy IAMXMON into DFHRPL Library

1. Copy the modules IAMXMON and IAMXMAP into a load library that is in the DFHRPL concatenation. IAMXMON is the Execution Monitor program and IAMXMAP is the mapset.

```
//COPYIAMX EXEC PGM=IEBCOPY,REGION=2M
//SYSPRINT DD SYSOUT=*  <- User Change
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(2))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(2))
//IAMLIB DD DISP=SHR,DSN=iam.loadlib  <- User Change
//DFHRPL DD DISP=SHR,DSN=cics.rpl
//SYSIN DD *
COPYMOD INDD=((IAMLIB,R)),OUTDD=DFHRPL
SELECT MEMBER=IAMXMON
SELECT MEMBER=IAMXMAP
/*
```

Figure 90-22: Example of JCL to copy IAMXMON to DFHRPL
PPT Entry for Mapset: IAMXMAP

2. Define new CICS Program Properties Table (PPT) entries for the Execution Monitor mapset which must be called IAMXMAP:

```
OBJECT CHARACTERISTICS
CEDA View Mapset( IAMXMAP )

Mapset        : IAMXMAP
Group         : IAMAPP1
Description   : 
REsident      : No | Yes
USAge         : Normal | Transient
USElpacopy    : No | Yes
Status        : Enabled | Disabled
RSl           : 00 | 0-24 | Public
```

Figure 90-23: Example of Defining PPT for the mapset: IAMXMAP

PPT Entry for Program: IAMXMON

3. Define a new CICS Program Properties Table (PPT) entry for the Execution Monitor program which must be call IAMXMON:

```
OBJECT CHARACTERISTICS
CEDA View PROGRAM( IAMXMON )
Program       : IAMXMON
Group         : IAMAPP1
Description   : 
Language      : Assembler | CObol | C | PLi | RPG
REload        : No | Yes
REsident      : No | Yes
USAge         : Normal | Transient
USElpacopy    : No | Yes
Status        : Enabled | Disabled
RSl           : 00 | 0-24 | Public
Cedf          : Yes | No
Datalocation  : Any | Below | Any
EXECKey       : Cics | User | Cics ← Note 1
REMOTE ATTRIBUTES
REMOTE System :
REMOTE Name   :
USElpacopy    : No | Yes
Transid       :
EXECutionset  : Fullapi | Dplsubset
```

Figure 90-24: Define PPT Entry for program: IAMXMON

**Note 1**: If CICS Storage Protection is active, the IAMXMON program **MUST be defined with EXECKey(CICS).**
4. Define a new CICS Program Control Table (PCT) entry defining the Transaction ID (TRANSID) you will use to invoke the IAM Execution Monitor. The selected TRANSID can be any unique 1 to 4 character identifier.

<table>
<thead>
<tr>
<th>OBJECT CHARACTERISTICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEDA View TRANSACTION( IAMX )</td>
</tr>
<tr>
<td>TRANSACTION : IAMX</td>
</tr>
<tr>
<td>Group : IAMAPP1</td>
</tr>
<tr>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>PROGRAM : IAMXMON</td>
</tr>
<tr>
<td>TRANSID : 00000 0-32767</td>
</tr>
<tr>
<td>PROFILE : DFHCICST</td>
</tr>
<tr>
<td>PARTITIONSET</td>
</tr>
<tr>
<td>STATUS : Enabled</td>
</tr>
<tr>
<td>PRIMESIZE : 00000 0-65520</td>
</tr>
<tr>
<td>TASKDATALOC : Any</td>
</tr>
<tr>
<td>TASKDATAKEY : Cics User</td>
</tr>
<tr>
<td>STORAGECLEAR : No</td>
</tr>
<tr>
<td>RUNAWAY : System</td>
</tr>
<tr>
<td>SHUTDOWN : Disabled Disabled</td>
</tr>
<tr>
<td>ISOLATE : Yes</td>
</tr>
<tr>
<td>REMOTE ATTRIBUTES</td>
</tr>
<tr>
<td>DYNAMIC : No</td>
</tr>
<tr>
<td>REMOTESYSTEM</td>
</tr>
<tr>
<td>REMOTENAME</td>
</tr>
<tr>
<td>TRPROF</td>
</tr>
<tr>
<td>LOCALQ : No</td>
</tr>
<tr>
<td>SCHEDULING</td>
</tr>
<tr>
<td>PRIORITY : 001 0-255</td>
</tr>
<tr>
<td>TCCLASS : No</td>
</tr>
<tr>
<td>TRANCLASS : DFHTCL00</td>
</tr>
<tr>
<td>ALIASES</td>
</tr>
<tr>
<td>ALIAS</td>
</tr>
<tr>
<td>TASKREQ</td>
</tr>
<tr>
<td>XTRANID</td>
</tr>
<tr>
<td>TPNAMESPACE</td>
</tr>
<tr>
<td>XTPNAME</td>
</tr>
<tr>
<td>RECOVERY</td>
</tr>
<tr>
<td>DTIMEOUT : No</td>
</tr>
<tr>
<td>INDOUT : Backout Backout</td>
</tr>
<tr>
<td>RESTART : No</td>
</tr>
<tr>
<td>SPURGE : No</td>
</tr>
<tr>
<td>TPUERGE : No</td>
</tr>
<tr>
<td>DUMP : Yes</td>
</tr>
<tr>
<td>TRACE : Yes</td>
</tr>
<tr>
<td>SECURITY</td>
</tr>
<tr>
<td>RESSEC : No</td>
</tr>
<tr>
<td>CMDSEC : No</td>
</tr>
<tr>
<td>EXTSEC : No</td>
</tr>
<tr>
<td>TRANSEC : 01 1-64</td>
</tr>
<tr>
<td>RSI : 00 0-24</td>
</tr>
</tbody>
</table>

Figure 90-25: Define PCT Entry for IAMXMON Transaction

Note 2: If CICS Storage Protection is active, the transaction MUST be defined with TASKDATAKEY(CICS).
Activating the IAM VSAM Interface

IAM provides a system level VSAM interface that provides the capability to use IAM datasets in place of VSAM KSDS, ESDS, or with the optional feature, AIX and RRDS files generally without any JCL or program changes. To accomplish this, IAM must be activated after each IPL. The activation procedure runs as a batch job or started task that terminates after initializing the IAM interface. Alternatively, if you are using IAM RLS, it will automatically activate the IAM VSAM interface during initialization. This activation effectively loads the IAM VSAM interface modules into the Pageable Link Pack Area (PLPA) area of virtual storage. The IAM VSAM interface modules act as a front end to various VSAM services, such as Open and Close. If the dataset being processed is an IAM dataset, then the interface gives control to the appropriate IAM routines. If the dataset is not an IAM dataset then control is directed to the normal VSAM processing routine. Once a dataset is opened, I/O requests are handled directly by the access method responsible for the dataset, either IAM or VSAM. The IAM VSAM interface causes no interference or overhead with the processing of I/O requests to real VSAM files.

- IAM IS NOT A REPLACEMENT FOR VSAM IN YOUR SYSTEM.
- IAM DOES NOT AFFECT THE NORMAL USE OF VSAM IN THE SYSTEM.
- VSAM CATALOG MANAGEMENT IS NOT CHANGED.

Coexistence with Other Software Products

There are wide varieties of other software products that also intercept the various services done for VSAM datasets. IAM can coexist in systems with these other software products installed. Depending on the function of the product and the various interfaces used, there may be a start order dependency between the different products to insure that the expected processing by each product occurs as anticipated. In general, if a product provides services and capabilities not applicable to IAM datasets, then they should be activated before IAM. Products that can be used for IAM datasets should be started after IAM. For example, VSAM buffering and performance products should be started before IAM. Other products, such as VSAM recovery and journaling software or VSAM space management packages that can be used for IAM datasets should be started after IAM.

There are some software products which require a specific setting in the IAM Global Options Table. These are identified in Section 90.10 of this manual and are shown in the table below.

Should you have any concerns or questions about IAM coexistence with other software products, contact the IAM technical support department at Innovation Data Processing.

ACF Security

Customers that are using the ACF product for security need to add the following definitions to their ACF parameters, to insure proper security for IAM files:

SET C(GSO)

INSERT SAFDEF.IAM MODE(GLOBAL) ID(IAM) RB(SVC019) RACROUTE(REQUEST=AUTH CLASS=DATASET REQSTOR=IAMAVSOC) REP

Top Secret

Users of Top Secret, and possibly other security software, will need to specify CREATE authority on any jobs, CICS, IAM/RLS, or IAM/PLEX address space that may to extend an IAM dataset on to a new secondary volume.
Activating the IAM VSAM Interface

IAM utilizes a major QNAME of IAMENQ to provide share option and other ENQ protection across multiple systems. Such ENQ's are issued with the specification of SYSTEMS for scope. Some products that handle the ENQ/DEQ for resources may need to have their parameters specified to make sure that IAMENQ is recognized as being a valid name for sharing resources between multiple systems. This is known to be necessary for CA-MIM, and may be needed for other products as well. An example of what needs to be specified for CA-MIM is shown below:

```
IAMENQ GDIF=YES, /* GDIF SHOULD PROCESS THIS QNAME
SCOPE=SYSTEMS,  /* GDIF TO PROCESS SYSTEMS ENQS
EXEMPT=NO,       /* DO NOT APPLY EXEMPT LIST
ECMF=YES,        /* ECMF REPORTS CONFLICTS
RPTAFTER=10,     /* REPORT CONFLICT AFTER 10 SECOND
RPTCYCLE=60,     /* AND EVERY 60 SECONDS THEREAFT
TRACE=NONE       /* NO TRACING
```

Figure 90-26: Control information required for CA-MIM with IAM

The start order recommendations of various products as they relate to IAM along with an indication of there being an IAM Global Option setting required are shown in the table below. Care must also be taken to insure that the various products are not starting at the same time because not all products follow the required locking protocol, and for some of the control block updates, there is no established serialization protocol. The order of the various start commands in the SYS1.PARMLIB member COMM1Dxx is no guarantee of the order in which the product will actually perform it's initialization since the commands are executed concurrently.

```
<table>
<thead>
<tr>
<th>Product</th>
<th>Start Order</th>
<th>Global Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIM Products</td>
<td>Before IAM</td>
<td>ENABLE=BIM</td>
</tr>
<tr>
<td>CA-DISK</td>
<td>After IAM</td>
<td>n/a</td>
</tr>
<tr>
<td>HYPERBUF</td>
<td>If started Before IAM</td>
<td>Zap C-90.0718</td>
</tr>
<tr>
<td>HYPERCACHE</td>
<td>Before IAM</td>
<td>n/a</td>
</tr>
<tr>
<td>BMC MAINVIEW</td>
<td></td>
<td>DYNCAT=YES</td>
</tr>
<tr>
<td>PROSMS</td>
<td>Before IAM</td>
<td>n/a</td>
</tr>
<tr>
<td>Recovery Plus</td>
<td>After IAM</td>
<td>n/a</td>
</tr>
<tr>
<td>CA-Allocate (SAMS (VAM))</td>
<td>After IAM</td>
<td>ENABLE=VAM</td>
</tr>
<tr>
<td>SmartBatch (IBM)</td>
<td>Before IAM</td>
<td>n/a</td>
</tr>
<tr>
<td>Transaction Server</td>
<td>After IAM</td>
<td>LOADABWO=YES</td>
</tr>
</tbody>
</table>
```

Figure 90-27: VIF Start Order with Other Software Products

The best way to insure the proper start order is by the use of automated operations software. Such software can determine that a process is complete by such means as checking for a specific WTO message, before initiating another process. For IAM the message to check for is the IAMW81 THE IAM SYSTEM MODULES ARE ACTIVE.

Another alternative is to start a multiple step PROC, which includes the IAMSTART procedure at the appropriate place. This will work providing that the steps prior to the IAMSTART terminate after the product initialization, as IAMSTART does.
Activating IAM

The IAM Installation Control Library contains seven (7) members to control the IAM VSAM Interface in your system. If you are running IAM RLS, you can let IAM RLS activate the IAM VSAM interface, instead of using the IAMSTART proc.

### MEMBER NAME | DESCRIPTION
--- | ---
VIFSTART | A job stream to activate IAM.
VIFSTATS | A job stream to check the status of IAM in the system.
VIFSTOP | A job stream to deactivate IAM.
VIFTEST | A job stream to activate a TEST IAM.
IAMSTART | A procedure that activates IAM.
IAMRLS | A procedure that starts IAM RLS and activates IAM (See Sec. 90.15)
IAMCOMM | A command to activate IAM at IPL time.

*Figure 90-28: ICL Members to Control the IAM VSAM Interface*

**VIFSTART Job**

The VIFSTART job can be used to activate the IAM VSAM Interface. This job requires both a STEPLIB and a SYSLIB DD statement, which specify the APF authorized IAM load library. If you already have a version of IAM active, refer to the section on Concurrent IAM and Test IAM below before starting the new version. Before executing this job, if you have any of the software products that are to be started after the IAM that are currently active, shut them down temporarily before starting IAM. After starting IAM, those software products can be restarted.

```bash
//VIFSTART JOB (IAM), 'IAM-VIF'
******
/* THIS JOB HAS BEEN CREATED FOR IAM V9.2 */
/* IT ACTIVATES THE IAM SYSTEM LEVEL VSAM INTERFACE */
/* THE PURPOSE OF THIS JOB IS TO INSTALL THE IAM SYSTEM */
/* LEVEL VSAM INTERFACE IN AN OS/390 or z/OS SYSTEM. */
/* USER CHANGES: */
/* CHANGE 'USER.LIB' TO NAME OF IAM LOAD LIBRARY */
******
//START EXEC PGM=IAMSTART
//STEPLIB DD DISP=SHR, DSN=USER.LIB ← USER CHANGE
//SYSLIB DD DISP=SHR, DSN=USER.LIB ← USER CHANGE
//SYSUDUMP DD SYSOUT=* 
```

*Figure 90-29: Example of the VIFSTART Job from the ICL*
**VIFSTATS Job**

The VIFSTATS job can be used to verify the status of the IAM VSAM Interface on your system. If you have multiple versions of IAM, always run the VIFSTATS with a STEPLIB to the most recent version. This is because older versions of IAM may not be able to properly display information on the newer versions. This job will provide WTO messages indicating the status of all levels of the IAM VSAM Interface on your system. The JOBLOG also includes messages indicating the names of the various modules IAM has placed in the PLPA, along with their entry point and level. If you have installed the IAM ISPF panels, this same information can be displayed on your terminal using the U.V options from the IAM primary panel.

```plaintext
//VIFSTATS JOB (IAM),'IAM-VIF'
******/
//*    THIS JOB HAS BEEN CREATED FOR IAM V9.2
//*    IT REPORTS ON THE STATUS OF THE IAM SYSTEM LEVEL
//*    VSAM INTERFACE.
//*    USER CHANGES:
//*    CHANGE 'USER.LIB' TO NAME OF IAM LOAD LIBRARY
******/
//STATS     EXEC PGM=IAMSTATS
//STEPLIB DD DISP=SHR,DSN=USER.LIB ← USER CHANGE
//SYSLIB DD DISP=SHR,DSN=USER.LIB ← USER CHANGE
//SYSUDUMP DD SYSOUT=* //
```

*Figure 90-30: Example of VIFSTATS Job from the IAM ICL*
Deactivating IAM

The IAM VSAM Interface can be deactivated with the VIFSTOP job. This job will not remove the IAM VSAM Interface modules from the PLPA, but will deactivate them. Should you absolutely need to remove the IAM modules from PLPA without doing an IPL, contact Innovation Data Processing for the procedure to follow.

NOTE: To reactivate IAM use the VIFSTART procedure.

WARNING: If you deactivate IAM in a production environment, all OPEN, CLOSE and IDCAMS processing against IAM files will fail.

VIFSTOP Job

```
//VIFSTOP JOB (IAM), 'IAM-VIF'
******
//* THIS JOB HAS BEEN CREATED FOR IAM V9.2
//* IT DEACTIVATE THE IAM SYSTEM LEVEL VSAM INTERFACE
//* USER CHANGES:
//* CHANGE 'USER.LIB' TO NAME OF IAM LOAD LIBRARY
******
//STOP   EXEC PGM=IAMPARE
//STEPLIB DD DISP=SHR, DSN=USER.LIB ← USER CHANGE
//SYSLIB  DD DISP=SHR, DSN=USER.LIB ← USER CHANGE
//SYSUDUMP DD SYSOUT=*
```

Figure 90-31: Figure 15: Example of VIFSTOP Job from the ICL
Also provided in the ICL is an example of a PROC that can be copied into a system PROCLIB, so that the IAM VSAM Interface can be automatically or manually started by the operator START command. This PROC is set up with one operand, LIB= that can be used to override the default STEPLIB and SYSLIB DD statements, which are required to be in the PROC. The STEPLIB can be removed if the PROC is only used to start the version of IAM that is in the system LINKLIST. If you have a need to be able to activate IAM without any DD cards, contact Innovation Data Processing for custom zap C-92.0009 to enable IAM starting directly from the LINKLIST if no SYSLIB DD is allocated.

```
//IAMSTART PROC LIB='USER.LIB' ← USER CHANGE
//* MEMBER(IAMSTART)
//*****
//* THIS PROCEDURE HAS BEEN CREATED FOR IAM V9.2
//* IT ACTIVATES THE IAM SYSTEM LEVEL VSAM INTERFACE.
//* THE PURPOSE OF THIS PROCEDURE IS TO PROVIDE A MEMBER THAT
//* CAN BE PLACED IN SYS1.PROCLIB THAT WILL ACTIVATE THE
//* INTERFACE WHEN IT IS INVOKED BY A SYSTEM START COMMAND.
//* USER CHANGES:
//* CHANGE 'USER.LIB' TO NAME OF IAM LOAD LIBRARY
//*****
//START EXEC PGM=IAMSTART
//STEPLIB DD DISP=SHR,DSN=&LIB
//SYSLIB DD DISP=SHR,DSN=&LIB
//SYSUDUMP DD SYSOUT=* 
```

Figure 90-32: Example of IAMSTART PROC from the IAM ICL

Activating IAM at IPL Time

After you have completed your testing, or possibly before, you will want to start IAM automatically at each IPL. To have the IAM automatically activated with each IPL, use the following procedure:

1. In member IAMSTART of the Installation Control Library, change the library name specified in the 'LIB=' statement to reflect your IAM load library.
2. Copy the member ‘IAMSTART’ into SYS1.PROCLIB (or the procedure library used by your installation for system level user procs). This is the start up procedure for IAM.
3. Copy the member IAMCOMM from the IAM Installation Control Library into the SYS1.PARMLIB member COMMND00 (or the COMMNDxx member used by your installation). This will create a record that will result in a system start command being issued for the IAM start up procedure at IPL time. The record in that member should look like the following statement:

   COM='START IAMSTART'

You have now completed the installation of a dynamic IAM system level VSAM interface that will automatically be activated each time the operating system is IPL'd.
If you are a current customer of IAM, you can install and test new versions of IAM without affecting any of your existing jobs. Programs can continue to use the prior version(s) until you choose to have them use the new version. Follow the appropriate testing procedure described below. Normally, your production version of IAM will be in a LINKLIST library. The new test version should be installed in a separate APF authorized library. If the library containing the new version is in the LINKLIST, it must be after the current version while testing is in progress.

One consideration with testing multiple levels of IAM is that the only version that can be used to define IAM datasets in JCL is the version of IAM that is in the LINKLIST. This is because the JOBLIB or STEPLIB specified in the JCL for the job to be executed is not used by the system initiator when it is performing allocation of the dataset.

If you are installing a new version of IAM, such as Version 9.2 while Version 9.1 (or earlier version) is your production IAM, you can run multiple levels of IAM for your testing. The major rule to be followed is that the oldest version of IAM must be started first. There are two different ways to run multiple levels of IAM. The first way is called Concurrent IAM, and is generally used when testing new versions of the product. The second way is called Test IAM, and is normally used when testing different maintenance levels of the same version of IAM. When running multiple versions of IAM, the primary production version matches the version of IAM that is in the system LINKLIST. The other versions are accessed through a JOBLIB or STEPLIB pointing to the alternate IAM load library.

Concurrent IAM works by comparing the loaded IAM module level with the level of the IAM VSAM interface using an internal level identifier for the IAM VSAM interface. When a VSAM function is intercepted by IAM, such as DEFINE, LISTCAT, OPEN, or CLOSE, IAM will load the appropriate IAM routine to process the request. If the internal level in the loaded IAM routine matches the IAM currently processing the request, then processing will proceed. If the internal level is not the same, then the current IAM percolates control to the prior level of IAM. If no match is found, VSAM will attempt to process the file, which will most likely result in a failure.

The Concurrent IAM VSAM can be started using the VIFSTART job or the IAMSTART PROC pointing to the new IAM load library. The one restriction is that the oldest version of IAM must be started first, and the newest version started last. Control over which version of IAM will be used is then based on the JOBLIB or STEPLIB being used by the job step. With Concurrent IAM, there is no concern about the job names, as there is with the Test IAM procedure. IAM V9.2 and above can be concurrently run with any prior levels of IAM. Please note that running IAM Version 9.2 with IAM Version 8.1 will cause the support for IAM files to use the XTIOT, Uncaptured UCB's, and DSAB above the line to be inactive.
Test IAM

IAM provides the capability to have a TEST version of the IAM VSAM Interface active. The TEST version can be used to test new versions and new maintenance levels of IAM, while having prior versions or maintenance levels of the IAM active. The TEST IAM interface utilizes a jobname screening mechanism to determine eligibility for using the TEST version. When activating a TEST version of IAM, a PARM of TEST is specified, followed by JOBNAME=jobname. The jobname must match the specified value to utilize the TEST version. The jobs that are utilizing the TEST version of IAM MUST HAVE A JOBLIB OR STEPLIB to the new IAM load library. A jobname prefix mask can be specified, by coding the prefix followed by an *. For example, specifying JOBNAME=myid*, all jobs that begin with the literal "myid" will be qualified for the test version. Up to five different jobnames or jobname groups can be specified. Multiple job names are specified by enclosing the names within parenthesis. The jobname(s) can be easily modified by rerunning the startup of the TEST version with a different value(s) for the JOBNAME parameter. Specifying JOBNAME=* will make all jobs eligible for the TEST version.

VIFTEST Job

In the IAM ICL, there is an example job stream to start a TEST version of IAM, as member VIFTEST. The VIFTEST JCL is as follows:

```plaintext
//VIFTEST JOB (IAM), 'IAM-VIF'
******
/*  THIS JOB HAS BEEN CREATED FOR IAM V9.2 */
/*  IT ACTIVATES THE IAM SYSTEM LEVEL VSAM INTERFACE */
/*  THE PURPOSE OF THIS JOB IS TO INSTALL THE IAM SYSTEM */
/*  LEVEL VSAM INTERFACE IN AN Z/OS SYSTEM */
/*  FOR TESTING PURPOSES. */
/*  USER CHANGES: */
/*  CHANGE 'USER.LIB' TO NAME OF IAM LOAD LIBRARY */
/*  JOBNAME=? TO JOBNAME= */
/*  THE JOBNAME TO BE TESTED, OR */
/*  TO [PREFIX]* FOR JOB NAME PREFIX, OR */
/*  * FOR ALL JOBS, OR */
/*  JOBNAME=(jobname1,jobname2,...,jobname5) */
/*  FOR MULTIPLE JOBNAMES */
******
//STARTTEST EXEC PGM=IAMSTART,PARM='TEST,JOBNAME=?' */
//STEPLIB DD DISP=SHR,DSN=USER.LIB ← USER CHANGE */
//SYSLIB DD DISP=SHR,DSN=USER.LIB ← USER CHANGE */
//SYSUDUMP DD SYSOUT=* */
//
Figure 90-33: Example of Starting a TEST VIF
```

As stated above, to change the name of the job(s) eligible for the TEST IAM version, simply rerun the above job with a new value specified for JOBNAME=.
Deactivating Test IAM

To deactivate the TEST IAM version, use the following JCL:

```
//VIFSTOPT JOB (IAM), 'IAM-VIF'
//STARTTEST EXEC PGM=IAMPARE, PARM='TEST'
//STAPLIB DD DISP=SHR, DSN=USER.LIB ↩ USER CHANGE
//SYSLIB DD DISP=SHR, DSN=USER.LIB ↩ USER CHANGE
//SYSUDUMP DD SYSOUT=*
```

*Figure 90-34: Example of JCL to Stop a Test VIF*

Test IAM Considerations

There are a few considerations when using a TEST IAM. The TEST IAM capability merely screens which job(s) are eligible for being processed by the new IAM. To be a valid test, the job must have access to the new IAM load modules via JOBLIB or a STEPLIB. The TEST IAM does not have the capability to dynamically select or force the library from which the IAM modules are loaded. This can cause failures or unexpected results if this capability is not appropriately used.

If the other version(s) of IAM that are active have different internal level values, then the TEST IAM behaves like a Concurrent IAM, with the exception that only those jobs whose name matches the specified JOBNAME value will be considered for the TEST IAM. For example, if a job has a JOBLIB or STEPLIB with the new IAM load library, but does not have a jobname match, it will end up processing with the real VSAM code, which will most likely cause an IEC1611 error message. However, if there is a jobname match but the IAM modules that are accessed by that job are from the prior version, then the prior version of IAM will be used.

When using a TEST IAM to test a new maintenance level, where the internal level values are the same, the TEST IAM controls which version of the IAM will handle the job. For job(s) whose name matches the specified value, they will be handled by the new IAM. Job(s) whose names do not match the specified value will be processed by the other (production) IAM. It is the users responsibility to insure that the correct version of the IAM libraries are used by the TEST and non-TEST jobs. This is easily controlled if the production version is in the system LINKLIST, and the TEST version is accessed by JOBLIB or STEPLIB. If the test job name matches the job name specified for the TEST IAM, and the job has a JOBLIB or STEPLIB for the new library, it will be processed entirely by the new version. If a job name does match the TEST VIF parameter, but the job does not have the test JOBLIB or STEPLIB, it will then execute using the new VIF modules, but the old processing modules. If a job name does not match the TEST IAM parameter, but has the new JOBLIB or STEPLIB, it will execute using the old IAM VSAM Interface modules and the new processing modules. Either one of these sequences may cause an invalid test, or even fail to properly execute.
When testing a new version or maintenance level of IAM, it will generally be desirable to have the multiple levels of IAM started during IPL. This can be most easily accomplished by modifying the IAMSTART PROC to have multiple steps, each one starting a different version of IAM. Remember that the older versions of IAM must be started before the newer versions. Below is an example of the IAMSTART PROC modified to start up two concurrent versions of the IAM.

```assembler
//IAMSTART PROC LIB='SYS1.IAM91.LOADLIB'
//** MEMBER(IAMSTART)
//******
//** THIS PROCEDURE HAS BEEN CREATED TO START MULTIPLE
//** VERSIONS OF THE IAM VIF.
//** USER CHANGES:
//** CHANGE 'SYS1.IAM91.LOADLIB' TO NAME OF PRODUCTION IAM
//** CHANGE 'SYS1.IAM92.LOADLIB' TO NAME OF NEW IAM
//******
//******
//START91 EXEC PGM=IAMSTART
//STEPLIB DD DISP=SHR,DSN=&LIB
//SYSLIB DD DISP=SHR,DSN=&LIB
//SYSUDUMP DD SYSOUT=* 
//START92 EXEC PGM=IAMSTART
//STEPLIB DD DISP=SHR,DSN=SYS1.IAM92.LOADLIB
//SYSLIB DD DISP=SHR,DSN=SYS1.IAM92.LOADLIB
//STATS EXEC PGM=IAMSTATS
//STEPLIB DD DISP=SHR,DSN=SYS1.IAM92.LOADLIB
//SYSLIB DD DISP=SHR,DSN=SYS1.IAM92.LOADLIB
```

*Figure 90-35: Starting Multiple IAM Versions during IPL*
You can also start up multiple levels of IAM including one as a TEST IAM. Below is an example of a PROC to do that. Remember that if you need to change the value of the JOBNAME all you need to do is to rerun the test IAM start with different jobname(s) specified.

```plaintext
//IAMSTART PROC LIB='SYS1.IAM91.LOADLIB'
//* MEMBER(IAMSTART)
******
//* THIS PROCEDURE HAS BEEN CREATED TO START MULTIPLE
//* VERSIONS OF THE IAM VIF WITH ONE AS A TEST VERSION.
//* USER CHANGES:
//* CHANGE 'SYS1.IAM91.LOADLIB' TO NAME OF PRODUCTION IAM
//* CHANGE 'SYS1.IAM92.LOADLIB' TO NAME OF TEST IAM
//* CHANGE 'iamtst*' TO THE DESIRED JOB NAME PREFIX FOR THE
//* TEST VIF.
******
******
//IAMPROD EXEC PGM=IAMSTART
//STEPLIB DD DISP=SHR,DSN=&LIB
//SYSLIB DD DISP=SHR,DSN=&LIB
//SYSUDUMP DD SYSOUT=* 
//IAMTEST EXEC PGM=IAMSTART,
// PARM='TEST,JOBNAME=iamtst*'
//STEPLIB DD DISP=SHR,DSN=SYS1.IAM92.LOADLIB 
//SYSLIB DD DISP=SHR,DSN=SYS1.IAM92.LOADLIB
//STATS EXEC PGM=IAMSTATS 
//STEPLIB DD DISP=SHR,DSN=&LIB
//SYSLIB DD DISP=SHR,DSN=&LIB
```

Figure 90-36: Example of PROC to Start Multiple IAM Versions with one as a TEST IAM
Testing IAM

Once you have completed the above steps, including activating IAM you are now ready to test IAM.

To make a quick test to see if IAM is properly installed in the system submit the job 'TESTVIF' from the IAM Installation Control Library. This job will execute IDCAMS to DEFINE, REPRO and DELETE an IAM file. A LISTCAT is executed against the IAM file. If IAM is properly installed, LISTCAT will show the file as a non-VSAM file. The IAMPRINT DD statement will display the characteristics of the IAM file. You should also verify from the IAMINFO and IAMPRINT reports that any of the changed Global Options did in fact take effect.

If LISTCAT shows the file as VSAM, then the IAM VSAM Interface may not be properly installed. Review the Installation Instructions to make sure that all required tasks have been completed. If there are any questions, contact IAM Technical Support at Innovation Data Processing.

NOTE: Before submitting the test job, review the JCL and control statements, making whatever changes are appropriate.

SMF Analysis

You are now ready to test IAM in your system. If you are a new user, the first thing you should do is identify your most active VSAM clusters. To aid you in this function, IAM supplies you with a SMF Analysis program (IAMSMFVS). This program is documented in Section 40.01 of this manual. To use this program you must be collecting SMF type 64 and either SMF type 4 or 30 subtype 4 records.

A sample procedure is supplied as member 'IAMSMFVS' in the IAM Installation Control Library.

Review this procedure making any appropriate changes. You can select from your current SMF dataset or from history SMF tapes. It is recommended that you scan from one week to one month's worth of SMF history to get a true picture of your VSAM usage. Submit member 'IAMSMFVS' to display the VSAM files with the most activity in your system. The first report will display the top 100 VSAM files in descending EXCP order. The second report will display the VSAM files in dataset name sequence. Converting just a few of your most active VSAM files can give noticeable and immediate benefits to your CICS and batch systems.

NOTE: If IAMSMFVS gives a ‘IAM495 NO RECORDS MATCHED SELECTION CRITERIA’ message, it probably means that you are not collecting SMF type 64 records. Check SYS1.PARMLIB member SMFPRMxx to see what records SMF is collecting. If type 64 records are not collected, you must change this member. You must issue a SET SMF=xx command to activate the SMF change or wait for your next IPL. After you have collected at least one week's worth of data, run the IAMSMFVS program again.
Once you choose which VSAM files you wish to test with, use IDCAMS to DEFINE the IAM file(s). Take your existing IDCAMS procedure and add the parameter OWNER($IAM) to the DEFINE statement.

Next, use IDCAMS REPRO to copy the VSAM cluster to the IAM file. Run a LISTCAT ALL against the IAM file with an IAMPRINT DD statement specified. The IAMPRINT report will give you the number of tracks used by IAM in addition to all of the IAM characteristics. IAM will automatically release a portion, based on CA% Freespace, of the unused space allocated, if secondary allocation has been specified. Compare this to the VSAM cluster. Unfortunately, it is not easy to tell how much space VSAM really used. If the Data Component HI-USED-RBA is close to HI-ALLOC-RBA, the VSAM file is using most of its allocated space.

If you wish to calculate the space used by the VSAM cluster use the following calculation. Use the values found in the DATA component. Divide the HI-USED-RBA by CISIZE. Divide the result by CIs per CA. Multiply this value by the number of tracks per CA. This will yield the number of tracks used by the Data Component and imbedded index (if specified). Add in the tracks for the Index. This will give you the approximate number of tracks used by the VSAM cluster.

One of the easiest tests you can make is to compare an IDCAMS REPRO of a VSAM cluster and IAM file to tape. Many applications use REPRO to make backup copies of their VSAM files. When the jobs are completed, compare the wall clock time, CPU time (TCB and SRB) and EXCPs issued by each job. This will give you a comparison of the resources used to sequentially read an entire VSAM cluster versus using an IAM file.

You are now ready to run parallel tests of jobs using VSAM clusters compared to IAM file(s). Select an application to test, using the IAMSMFVS report as a guide. Use IDCAMS to DEFINE the IAM file specifying a different cluster name(s) then REPRO the VSAM cluster(s) into the IAM file(s). Use a copy of any file(s) that may be changed by the application. Run the production job against the VSAM cluster(s). Use the same JCL for the parallel run, changing the JCL to point to the new IAM file(s). Insert a STEPLIB pointing to the IAM load library if it is not in the LINKLIST. Re-execute the production run using the IAM file(s). Compare the results from each run. If you do not have the reporting tools necessary to get the SMF data you require, use the IAM supplied program IAMSMF (Section 41). Compare the statistics for each job. You will want to measure the wall clock time, CPU time (SRB and TCB), and EXCP counts (DATA and INDEX for VSAM). In addition, you will want to compare the disk space used by IAM versus the VSAM clusters.

Add the following statement to each of the steps using IAM data test.

```
//IAMINFO DD SYSOUT=*  
```

If this statement is present IAM will print a Run Time Statistics report each time an IAM file is closed. The report will display the characteristics of the file, run time statistics (memory use, read and write I/Os, etc.), count of each command issued (GETS, PUTS, ADDS, DELETES, POINTS, etc.), number of buffers used and additional information. There is very little overhead associated with the report because all the statistics are kept regardless of whether or not an IAMINFO DD card is present. The information from the IAMINFO report can be very valuable to you and Innovation in determining what each job is doing. (see Section 10.72 IAMINFO Reports for details on using this report.)
Testing a New Version of IAM

Testing a new version of IAM or a new maintenance level of IAM is straightforward. If you have activated a Concurrent IAM VSAM interface, then just add the appropriate STEPLIB or JOBLIB to the job steps or jobs that you want to run as a test with the new version or level of IAM. If you are using a TEST IAM, then the job name(s) must match the value specified when you last started the TEST IAM along with the appropriate STEPLIB or JOBLIB. By having IAMINFO reports and/or IAMPRINT reports, you can verify that the job is running with the new version or level from the heading on those reports.

CAUTION

If you plan on using the same IAM data sets with V8.1 or older version of IAM, make sure that you do not select Global Options or IAM Overrides that may cause the data set to be unusable under the prior version. Please note that Version 9.2 supports Prime Related Overflow format files and data sets on EAV volumes. Do not create IAM files in these formats if they might be used with a prior level of IAM.
IAM Installation Questions

Here are the answers to some common questions that may be encountered after the installation of IAM.

**Question:** Why do I get a VSAM cluster when I have placed the parameter OWNER($IAM) in the IDCAMS DEFINE?

**Answer:** There are several possibilities:

1. The IAM VSAM Interface is not active in this system. Run the procedure VIFSTATS in the Installation Control Library to see if IAM is active. If IAM is not active, run VIFSTART to activate VIF.
2. The IAM load library is not in the LINKLIST. The JCL LOG in this case may indicate a S806 abend for an IAM module name. If you do not wish to put IAM in the LINKLIST, insert a STEPLIB pointing to the IAM library in the IDCAMS DEFINE job.
3. The IAM library is not authorized. The IAM library must be authorized.

**Question:** Why do I get VSAM error message IEC161I opening an IAM file?

**Answer:** There are several possibilities:

1. The IAM VSAM Interface is not active in this system. Run the procedure VIFSTATS in the Installation Control Library to see if IAM is active. If IAM is not active run VIFSTART to activate VIF.
2. The IAM load library is not in the LINKLIST. If you do not wish to put IAM in the LINKLIST, insert a STEPLIB pointing to the IAM library in the job referencing the IAM file.
3. The IAM library is not authorized. The IAM library must be authorized.

**Question:** Why might a file I converted to IAM all of a sudden go back to a VSAM cluster?

**Answer:** Some of the possibilities are:

1. Make sure that the IAM VSAM interface is active on the system.
2. There was a job that deleted and re-defined the file, but did not have the required JOBLIB or STEPLIB.
3. The JOBLIB or STEPLIB was not authorized in the system’s APF list.
4. There was a job executed against the IAM file which DELETEd and re-DEFINEd the file as VSAM without the OWNER($IAM) parameter specified. You must find this job and change the DEFINE procedure. Use the IAMSMSF program copy and query commands to report on all the jobs that have defined the cluster so you can identify the IDCAMS job causing the problem.
**Question:** Can I test a new version of IAM if I am already using an older version of IAM?

**Answer:** Yes, please refer to the detailed instructions in Section 90.30.

**Question:** Can I use IAM with other software products that do VSAM allocation control or otherwise intercept VSAM?

**Answer:** Yes, IAM supports or coexists with the majority of the many software products that make decisions based on a file's DSORG and or interface with the same z/OS services as IAM. Please see Section 90.20 for start order dependences with various products.
Moving a New Version of IAM into Production

Overview

Several ways can be used to place the new version or maintenance release of IAM into production. In all cases, it will make for a cleaner and safer process if you can schedule an IPL. However, many installations are not able to do that, so alternative procedures are presented that will accomplish activating the new version or maintenance release as the level used by all of the jobs in the system without an IPL. Be sure to review the IPL implementation procedure because there are tasks you will need to do so that the new version of IAM will remain in place after an IPL occurs.

It is highly recommended that no matter what procedure you decide to follow, that you keep a copy of your prior version of IAM available in an authorized library. By doing so, you will have quick access to the prior version if it becomes necessary. The recommended procedures provide the capability of an emergency fall back to the prior version without an IPL, should unexpected problems occur. Once you have reached a level where the fall back is no longer necessary, you will be able to remove the old version of IAM from your system.

Implementation Strategy

During the testing phase of the new level of IAM, you have been running with two versions of IAM active, using JOBLIB or STEPLIB to access the new version, while running out of LINKLIST for the production version. The recommended strategy is to migrate to the new level of IAM in two phases. During the first production phase for the new version, there will still be two levels of IAM started. However, the new level of IAM will be placed in the LINKLIST, while the prior version of IAM will be accessible, if necessary, through a JOBLIB or STEPLIB. The old version of IAM, while started, will be deactivated. This will leave the prior level of IAM modules in PLPA, however they will be inactive. The old level of IAM will be ready to be reactivated if the need arises.

Once the new version has been completely accepted, then the second phase can be performed. This phase will remove the old level of IAM from the system completely. During this phase, changes will be made to the IPL procedure for activating IAM. The change will be to activate just the new production level of IAM. The old library, if it is still in the LINKLIST, can be removed. Then, after an IPL, the old IAM libraries can be deleted from the system.

In planning for the implementation, it is highly recommended that there be only one load library with any particular level of IAM. The primary reason for this recommendation is to avoid the problems encountered when there are multiple load libraries for the same version. The potential problems are differences in IAM Global Options settings, and maintenance concerns when applying zaps. If you do have to copy the IAM product from one library to another, be sure to delete the original library as soon as possible to prevent inadvertent use. Also, be sure to copy all of the IAM load modules. There are occasionally new modules added to the IAM load library, and there may be some modules eliminated.

Preparation for Production

The method of implementation for the first phase that will be least likely to cause problems is to perform an IPL to put the new version of IAM into production. Because many installations are not able to use this technique, there are alternative procedures provided below to implement the new version of IAM without an IPL. However, many of the tasks described below will have to be done even with the alternative procedures, so please be sure to review this methodology.
There are two steps to prepare for the implementation. First, is to revise the startup procedure(s) for IAM and second is to change the LINKLIST structure to include the new level of IAM. The changes to the startup procedure for IAM include adding a step to deactivate the old level of IAM after it is started, and if a TEST IAM is being used, change the JOBNAME value to JOBNAME=*, so that all jobs are eligible for the new version. Shown below is the revised multiple step PROC for starting Concurrent IAM, with a deactivation step inserted for the old version.

```
//IAMSTART PROC LIB='SYS1.IAM91.LOADLIB'
//** MEMBER(IAMSTART)
//******
/// THIS PROCEDURE HAS BEEN CREATED TO START MULTIPLE
/// VERSIONS OF THE IAM VIF.
/// USER CHANGES:
/// CHANGE 'SYS1.IAM91.LOADLIB' TO NAME OF OLD VERSION OF IAM
/// CHANGE 'SYS1.IAM92.LOADLIB' TO NAME OF NEW VERSION OF IAM
//******
//********
//START91 EXEC PGM=IAMSTART
//STEPLIB DD DISP=SHR,DSN=LIB
//SYSLIB DD DISP=SHR,DSN=LIB
//SYSDUMP DD SYSOUT=*  ← Deactivate old version
//DEACT91 EXEC PGM=IAMPARE
//STEPLIB DD DISP=SHR,DSN=LIB
//SYSLIB DD DISP=SHR,DSN=LIB
//START92 EXEC PGM=IAMSTART
//STEPLIB DD DISP=SHR,DSN=SYS1.IAM92.LOADLIB
//SYSLIB DD DISP=SHR,DSN=SYS1.IAM92.LOADLIB
//STATS EXEC PGM=IAMSTATS
//STEPLIB DD DISP=SHR,DSN=LIB
//SYSLIB DD DISP=SHR,DSN=LIB
```

Figure 90-37: Example of PROC to Start Two IAM VSAM Interfaces and Deactivate One

The second step is to put the new IAM load library into the LINKLIST (in the appropriate LNKLISTxx or PROGxx member of SYS1.PARMLIB), and to optionally remove the old library from the LINKLIST. The alternative to changing SYS1.PARMLIB is to copy the new version of IAM into an existing LINKLIST library. That job should be run just prior to the IPL. If the plans are to copy the new IAM into the library where the old version resided, make sure that you have a copy of that version of IAM in another APF authorized library. If both the new and the old libraries are in LINKLIST, make sure that the new production version is in front of the old version.

After the IPL, the new version will be the production version. From the IAMSTATS, you will see that the old level of IAM is in storage, but INACTIVE. Should the need arise, the old version can be reactivated by running IAMSTART with that version's libraries as the STEPLIB and SYSLIB. Once the old version is reactivated, it can be accessed by JOBLIB or STEPLIB for selected jobs, or the library can be copied into LINKLIST in front of or over the new version, followed by an LLA refresh.
Moving a New Version of IAM into Production

z/OS has the capabilities to dynamically change the system LINKLIST. This facility will make it possible to put the new version of IAM into production with minimal impact to the ongoing operation of the system. This procedure can also be used to put a new maintenance level of IAM into production, however you should run some tests prior to doing the cut over with running the new level of IAM with the old level of IAM.

1. Make sure that each version of IAM is in it's own unique library, and that each corresponding version of the IAM VSAM interface has been activated in the proper sequence with any other potentially interacting software products.

2. If the new IAM is a TEST IAM, set the JOBNAME=* so that it will process all jobs.

3. Issue the following Operator Commands:
   - **SETPROG LNKLST,DEFINE,NAME=NEWIAM,COPYFROM=CURRENT**
     This will build a new LINKLIST based on the currently active LINKLIST.
   - **SETPROG LNKLST,ADD,NAME=NEWIAM,DSNAME=newiam.loadlib,ATTOP**
     This command will place the new IAM library at the top of the LINKLIST concatenation, after SYS1.LINKLIB. Alternatively, instead of ATTOP, you can specify AFTER=dsname to place the new IAM library in a particular location. Be sure that if you do use AFTER= that the new IAM library will still be in front of the old IAM library.
   - **SETPROG LNKLST,DELETE,NAME=NEWIAM,DSNAME=oldiam.loadlib**
     This command is optional, it will remove the old IAM library from the new LINKLIST.
   - **SETPROG LNKLST,ACTIVATE,NAME=NEWIAM**
     This will activate the new LINKLIST for all new jobs and address spaces. Currently executing jobs will continue to use the old version of IAM.

4. Prior to performing an IPL, make sure to update the SYS1.PARMLIB with the LINKLIST changes, as well as changing the IAMSTART PROC if necessary.

When using the above procedure, do not deactivate the old version of IAM until you are certain that all jobs that were using that version of IAM have terminated. If you need to back out the new level of IAM, you can either activate the original LINKLIST, if you know the name, or build a new LINKLIST as the commands above do, deleting the new IAM library, and adding the old IAM library if you had deleted it. If you were running a TEST IAM where the prior IAM had the same internal level identifier, then deactivate the TEST IAM with the VIFSTOP JCL.
Without Dynamic LINKLIST

The following procedure to put IAM into production without using the Dynamic Linklist facilities. Because this procedure does not utilize the dynamic LINKLIST, it will require a brief period of time where all activity against IAM files must be quiesced.

1. The first step is to make sure that there is an APF authorized backup library with the current production version of IAM. This must be separate from the target production LINKLIST library for the new version.

2. The second step is to make sure that both the current production version of IAM and the new version of IAM have been activated in the correct order regarding other software products in your installation. This is best accomplished by using one of the suggested multiple step IAM start procedures.
   - All activity on IAM files must be stopped. All open IAM files must be closed. When that point is reached, the following tasks are to be done:
     - Deactivate the production version of IAM using the VIFSTOP job.
     - Copy the new IAM library into the LINKLIST. The copy can be done into a library in front of the current production IAM library in LINKLIST, or into the current production library in LINKLIST. Make sure that the library that you copy the new IAM version into does not take any extents. If it does, a re-IPL will be necessary if running a compress does not get the dataset back down to its original size.
     - Refresh the LINKLIST with the F LLA,REFRESH command.
     - If the new version is running as a TEST IAM, set JOBNAME=* by running the VIF START job with a PARM='TEST,JOBNAME=*'.

3. The new IAM is ready for use. IAM file activity can be resumed with the new production version of IAM.

4. When a back out procedure is no longer needed, revise the IAM start procedure for IPL to activate only the new production level of IAM. Then whenever the system is re-IPL'd, the new production version will be the only version of IAM installed on the system, and the backup library for the old version can be deleted.

   If you need to backout the new IAM version, reactivate the old IAM, then copy the old IAM library back into the LINKLIST over the new version, and do an LLA Refresh. Once that is complete, you can deactivate the new IAM and resume processing of your jobs. Be sure to back out any changes you may have made to SYS1.PARMLIB for LINKLIST or the IAMSTART procedure.

Final Steps

The final phase of implementation of the new version or maintenance level of IAM can take place once you are satisfied with the performance and reliability of the product. To accomplish this task, change the start up of IAM during IPL to only start the new production version of IAM. If you are using the recommended multiple step PROC, then remove the start up and deactivation of the prior release. If the new IAM was being run as a TEST IAM, it can now be turned into a regular IAM by removing the parameter fields. (i.e. PARM='TEST,JOBNAME=**) The load library with the old IAM version can now be removed from the LINKLIST, and also from the APF list, presuming that the dataset was being used for IAM only. After the system is re-IPL'd, which can be done whenever it is convenient, then the new version of IAM is in full production. At this point, the old libraries can be deleted from your system.
91 IAM GLOBAL OPTION CHANGE FACILITY

91.01 IAMZAPOP – Overview and JCL

The Global Option Change Facility gives the user a simple method of modifying installation options for executing the IAM system. The installation options include defaults for setting certain file attributes during DEFINE, enabling special features in IAM, and other processing options. This facility supplies the user with the ability to determine what options they have changed with the AUDIT command, to PRINT the global option values, change values using the ZAP command, and RESET options to their original values as supplied on the installation tape. You should carefully review the options available for the IAM system.

Batch Execution

To execute IAMZAPOP as a BATCH job use the following JCL:

```
//IAMZAPOP EXEC PGM=IAMZAPOP
//STEPLIB DD DISP=SHR,DSN=iam.loadlib
//SYSLIB DD DISP=SHR,DSN=iam.loadlib
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *
  
pull control cards here
/
```

Figure 91-1: Example of JCL to Execute IAMZAPOP

EXEC PGM= Must specify the program name of the Global Option Change Facility - IAMZAPOP.

STEPLIB DD Be sure that you run with the IAMZAPOP version that matches the Global Options Table that you are modifying by specifying a STEPLIB to the same library as the SYSLIB.

SYSLIB DD Required DD that must specify the load module library in which the IAM Global Option Table resides.

SYSPRINT DD Specifies the output message data set. This is a required statement and usually is a SYSOUT data set.

COPYTODD DD Required DD only for the COPY operation, which specifies the load library containing the target option table for the COPY operation. The changed options in the SYSLIB option table will be copied to the target option table.

SYSIN DD Required DD, which specifies the control statement data set required for all functions. Usually an input stream or DD * data set.
TSO Execution

The GLOBAL OPTION CHANGE FACILITY program (IAMZAPOP) can be executed under TSO. The program will prompt the user for the commands. 'END' will save the new options, if any, and terminate the program. The allocations required to execute IAMZAPOP in the TSO Foreground are as follows:

```
ALLOC F(SYSLIB) DA('iam.library') SHR
ALLOC F(SYSPRINT) DA(*)
ALLOC F(SYSIN) DA(*)
IAMZAPOP

------- or if the IAM library is not in LINKLIST-------
CALL 'iam.library(IAMZAPOP)'
```

Figure 91-2: Invoking IAMZAPOP under TSO
IAMZAPOP – Format and Options

Available Functions

The Global Option Change Facility has the following commands:

AUDIT
Lists the Global Options that have been changed from their distributed defaults.

The AUDIT command format is: AUDIT

CANCEL
Terminates IAMZAPOP without updating the Option Table.

The CANCEL command format is: CANCEL

COPY
Copies the changed Global Options from the Option table in the SYSLIB dataset to the COPYTODD dataset. COPYTODD is a required DD statement for this operation.

WARNING: DO NOT COPY PRIOR VERSION GLOBAL OPTIONS INTO V9.2 LIBRARY. DOING SO WILL CAUSE V9.2 TO FAIL. INSTEAD DO AN AUDIT ON THE PRIOR VERSION AND THEN ZAP THE CHANGES IN THE NEW VERSION.

The COPY command format is: COPY

END
Terminates IAMZAPOP processing and rewrites the Option Table if any option was changed. This command is intended for TSO users.

The END command format is: END

HELP
The HELP command will print or display a menu of the IAMZAPOP options and related documentation.

The HELP command format is: HELP ALL

PRINT
If PRINT is specified, IAM will print or display the current values of the Global Options table.

The PRINT command format is: PRINT

RESET
If RESET is specified, IAM will reset the Global Options to the original values on the installation tape.

The RESET command format is: RESET

ZAP
Modify the Global Options. This command can be used to enable or disable specified processing options and set DEFINE defaults for the IAM system. The operands for this command are documented by function in the following sections.

The ZAP command format is: ZAP operand=value, ...operand=value
91.03 IAMZAPOP – ZAP Command

This section discusses options that apply to most of the programs within the IAM system.

ZAP  [BELOWPOOL = YES | NO ]  [,BUFOPNO=nn]
[ ,BUFSP=nnnnnnn]  [,CICSBUFSP=nnnnnnn]
[ ,CICSTID=cccc]  [,CORELIMIT=nnnnnnn]
[ ,COMPRESSTYPE=(HARDWARE | Hw | SOFTWARE | Sw)]
[ ,CRBUFOPT=(CYL | MCYL | MTRK | TRK | 64BIT )]
[ ,DATACOMPRESS=nnnnnnn]  [,DATASPACE=nnnn]
[ ,DESCRIPTCODE=nn]  
[ ,DISABLE=(BIM | BUF64 | ENHANCED | ESDSLOCK | EURODATE |
  EAV | HPF | LARGE | NOMEMLIMIT | NOOWNID | NOREUSE | PRO |
  PSEUDO | TURBO | REORGWTO | RLSJRNL | SAFRC4 | VAM | XESDS)]
[ ,DSORG= DA | PS ]  [,DYNCAT= YES | NO ]
[ ,ENABLE=(BIM | BUF64 | ENHANCED | ESDSLOCK | EURODATE |
  EAV | HPF | LARGE | NOMEMLIMIT | NOOWNID | NOREUSE | PRO |
  PSEUDO | TURBO | REORGWTO | RLSJRNL | SAFRC4 | VAM | XESDS)]
[ ,ESDSINTEGRATED=nn ]  [,IAMW22LIM=nnnnn]
[ ,IMMEDWRITE= NONE | ALL | RLS | CICS | BATCH]
[ ,INDEXSPACE= 64BIT | ALL | CICS | NO ]
[ ,LIMITKEYS=nn ]  [,LOADABWo= YES | NO]
[ ,LONGKEYS=nnnn]  
[ ,MAXBUFNO=nn ]  [,MAXOVERFLOW=nnnnnnn]
[ ,MAXREGION=nnnnn]  [,MAXSECONDARY=(x,y)]
[ ,MINCOMPRESS=nn ]  [,MULTIVOLUME= PRIMARY | SECONDARY]
[ ,OCOREO%=nnn]  [,OCOREX%=nnn]
[ ,PE=nnnnn ]  [,RECFM= F | V]
[ ,RECTYPE=nnn]  [,RELEASE= YES | NO ]
[ ,RLS={SHARE1|SHARE2|SHARE3|SHARE4},{AND|OR},
  {TABLE}| NONE , {OPTIONAL | REQUIRED}]
[ ,RLSID=cccc]
[ ,ROUTECODE=nn ]  [,SMF= YES | NO ]
[ ,SORTCORE=nnnnnnnn]  [,SORTMSG=xx]
[ ,SORTPPFX=xxxx]  [,STORCLASS=c......c]
[ ,VAROVERFLOW=YES|NO ]  [,VSAMBLOCKF=n]
[ ,VSAMwTO=YES|NO ]  [,WORKDDNAME=xxxxxxx]
[ ,WORKPRIMARY=nnn]  [,WORKSECONDARY=nnn]
[ ,WORKUNIT=c......c]
OPERANDS

Following are the operands for the ZAP command.

<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BELOWPOOL=</td>
<td>Specifies whether or not to use IAM below the line storage pools for I/O control blocks under CICS and IAM RLS. The default is to use a dedicated 4K below the line storage area for each open Enhanced format IAM file. For CICS or IAM RLS regions with many open Enhanced format IAM files opened concurrently, this feature should reduce the below the line storage requirements by at least 50%. Valid values are YES or NO. The default value is YES, which means that the storage pooling feature will be used.</td>
</tr>
<tr>
<td>BUFO</td>
<td>Specifies a default minimum value for the initial number of buffers IAM will acquire when opening a file for processing. For batch jobs, the default is the number of blocks on 16 tracks or the number of blocks in the data set if lower. For CICS, the number of blocks on 15 tracks or the number of blocks in data set if lower. The MINBUFN0 override will be used if it is larger. IAM's Real Time Tuning starts with this initial number of buffers and during file processing dynamically adjusts the number of buffers actually used for a file up or down based on demand. You may specify a value from 1 to 8192 buffers. The default is 4.</td>
</tr>
<tr>
<td>BUFSP</td>
<td>Specifies a default, in kilobytes, for the maximum amount of storage that IAM is to use for buffers when accessing a file in environments other than CICS. IAM divides this value by the file's block size to determine the number of buffers that will fit. For example: a file with 1/4 track blocking (13,682) on a 3390, IAM can fit 65 buffers in 875K of storage. IAM will use this value, or MAXBUFNO Global Option, which ever is higher, to set the maximum number of buffers for processing a dataset, unless overridden. BUFSP must be at least 65. The default is 32,768, which will yield a value that will hold approximately 40 cylinders of data blocks.</td>
</tr>
<tr>
<td>CICSBUFSP</td>
<td>Specifies a default, in kilobytes, for the maximum amount of storage that IAM is to use for buffers when accessing a file under CICS. IAM divides this value by the file's block size to determine the number of buffers that will fit. For example: When a file with 1/4 track blocking (13,682) on a 3390, IAM can fit 19 buffers in 256K of storage. IAM will use this value, or MAXBUFNO Global Option, which ever is higher, to set the maximum number of buffers for processing a dataset, unless overridden. CICSBUFSP must be at least 65. The default is 1024.</td>
</tr>
<tr>
<td>CICSTXID</td>
<td>Specifies the name of the IAM supplied CICS transaction that will handle the automatic disconnect and automatic reconnect function for IAM/RLS and IAM/PLEX. Default is $IAM.</td>
</tr>
<tr>
<td>Operand</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>COMPRESSTYPE</td>
<td>Specifies the default compression algorithm to use.</td>
</tr>
<tr>
<td></td>
<td><strong>SOFTWARE</strong> or <strong>SW</strong> – Use IAM’s proprietary software compression.</td>
</tr>
<tr>
<td></td>
<td><strong>HARDWARE</strong> or <strong>HW</strong> – Use the IBM hardware compression instruction.</td>
</tr>
<tr>
<td></td>
<td>The default is <strong>Software</strong>.</td>
</tr>
<tr>
<td>CORELIMIT</td>
<td>Specifies the minimum prime index size required for IAM to consider using a compressed index structure. Any number from 0 to 999999, inclusive, may be specified. The default is <strong>8000 bytes</strong>.</td>
</tr>
<tr>
<td>CRBUFOPT</td>
<td>Specifies the EXCP buffer option to be used during a file load process. The valid values are:</td>
</tr>
<tr>
<td></td>
<td><strong>CYL</strong> – Acquire enough buffers for one full cylinder. Each physical I/O (EXCP) is for one half of a cylinder.</td>
</tr>
<tr>
<td></td>
<td><strong>MCYL</strong> – Acquire enough buffers for two full cylinders. Each physical I/O (EXCP) is for a full cylinder.</td>
</tr>
<tr>
<td></td>
<td><strong>TRK</strong> – Acquire enough buffers for two tracks. Each physical I/O (EXCP) is for one track.</td>
</tr>
<tr>
<td></td>
<td><strong>MTRK</strong> – Acquire enough buffers for ten tracks. Each physical I/O (EXCP) is for five tracks.</td>
</tr>
<tr>
<td></td>
<td><strong>64BIT</strong> – Similar to <strong>MCYL</strong> except that storage is acquired 64-bit addressable storage.</td>
</tr>
<tr>
<td></td>
<td>Default value is <strong>MCYL</strong>.</td>
</tr>
<tr>
<td>DATACOMPRESS</td>
<td>Specifies the smallest size IAM file that will be considered for automatic data compression. This value is the minimum number of tracks that a DEFINE can specify as a file’s primary allocation and still qualify for automatic data compression. Any number from 0 to 99999999, inclusive may be specified. The default is <strong>75</strong>. IAM data sets that are defined as being 75 tracks or larger will default to being data compressed.</td>
</tr>
<tr>
<td>DATASPACE</td>
<td>Specifies the size, in megabytes, of the Data Space to be used for the temporary storage of the index to the IAM file that is being loaded. IAM will also use this value for the Index Space size, with a maximum Index space size of four times this value. Valid values are from 0 to 2048. A value of 0 results in the use of a dynamically allocated temporary data set. Default is <strong>2048 megabytes</strong>.</td>
</tr>
<tr>
<td>DEScriptCODE</td>
<td>Specifies the descriptor code(s) to be used when issuing Write-To-Operator or Write-To-Operator-With-Reply messages. Any number from 1 to 16 inclusive may be specified. Multiple descriptor codes can be entered if specified as <strong>DESCRIPTCODE=(nn,...,nn)</strong>. The default is <strong>0 (X'0000')</strong>.</td>
</tr>
<tr>
<td>DSORG</td>
<td>Specifies the DSORG to be used when creating an IAM file. Innovation recommends and highly encourages users to retain the default value of PS.</td>
</tr>
<tr>
<td></td>
<td><strong>DA</strong> – Sets a DSORG of DA (direct access) – Not Recommended</td>
</tr>
<tr>
<td></td>
<td><strong>PS</strong> – Sets a DSORG of PS (physical sequential)</td>
</tr>
<tr>
<td></td>
<td>The default is <strong>PS</strong>.</td>
</tr>
</tbody>
</table>
**Operand** | **Description**
--- | ---
**DYNCAT** | Specifies whether IAM should allow dynamic allocation to catalog the IAM file. This option is available to enhance IAM's support of BMC MAINVIEW (formerly POOLDASD). Valid values are YES, let dynamic allocation catalog file or NO in which case IAM will catalog the file. **The default is NO.**
**ENABLE=** or **DISABLE=** | The option(s) coded for this operand shall be activated.
**BIM** | Enables IAM support for the BIM product. **Default value is disabled.** Requires a restart of IAM to take effect.
**BUF64** | Specifies that storage for the buffers for read and update processing be acquired in 64-bit virtual storage (above the bar). **Default is that this option is disabled** due to potential for increased CPU time when performing I/O.
**EAV** | When enabled, specifies that IAM files will automatically be defined as eligible to use the extended area on EAV volumes. This is functionally setting EATTR=OPT on all IAM dataset allocation. It can be overridden by explicitly specifying EATTR(NO) on the IDCAMS Define Cluster control cards. **Default is that this option is disabled.**
**ENHANCED** | Specifies that IAM files will default to the Enhanced file format when they are defined. **Default is that this option is enabled.**
**ESDSLOCK** | A new option to control the CI lock obtained on the high used RBA in an ESDS file while adding records. If many applications are adding records to an ESDS file and the applications are dependent on the order in which these adds are done, this option should be enabled. An exclusive control reason code will be returned to the application if the CI is locked and it can reissue the request. If the order of adds is not important then this option can be disabled for better performance. **Default is that this option is enabled.**
**EURODATE** | Changes the format of date fields on IAMINFO and IAMPRINT reports to a European format of dd/mm/yyyy instead of mm/dd/yyyy. **Default is disabled.**
**HPF** | Allows IAM to utilize z/HPF type of channel programs when z/OS indicates it is supported on the device(s) that the dataset resides on. **Default is enabled.** This can be disabled if the use of z/HPF channel programs is causing any problems.
**LARGE** | When enabled specifies that the IAM dataset will default to using the DSNTYPE of LARGE. Innovation recommends that this format be used instead of SMS Extended Format. Enabling this option will override any DSNTYPE specified in a Data Class. It can be overridden by using the DSNTYPE= IAM CREATE override when defining the dataset. **Default is that this option is disabled.**
**NOMEMLIMIT** | When enabled, will indicate that IAM may exceed the 64-bit virtual storage MEMLIMIT when necessary on storage requests for the index. **Default is disabled** meaning MEMLIMIT is honored. This option is provided to avoid failures or in case a failure occurs due to insufficient storage in 64-bit virtual for the index.
**NOOWNID** | Causes IAM to save the actual value specified for OWNER on the define in the catalog entry for the OWNER field. **The default is Enabled.**
**Operand**  
**ENABLE or DISABLE** (continued...)  

<table>
<thead>
<tr>
<th><strong>Description</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NOREUSE – Causes IAM to honor the NOREUSE option if coded in the IDCAMS define statements. <strong>The default is Disabled</strong>, which sets all IAM data sets as reusable, except for IAM base clusters with an alternate index.</td>
<td></td>
</tr>
<tr>
<td>PRO – Will cause ENHANCED format files to be allocated using the new PRIME RELATED OVERFLOW format. <strong>The default is DISABLED</strong>.</td>
<td></td>
</tr>
<tr>
<td>PSEUDO – When enabled will cause IAM to use the PSEUDORBA format for IAM ESDS type of files. <strong>The default is DISABLED</strong>.</td>
<td></td>
</tr>
<tr>
<td>REORGWTO – Causes IAM to display the IAMW22 messages when appropriate. To eliminate those messages, DISABLE this value. <strong>The default is Enabled</strong>.</td>
<td></td>
</tr>
<tr>
<td>RLSJRNL – When enabled this option specifies that IAM will use the IAM RLS journals for those data sets that have IAM journaling specified via the JRNAD override or the IDCAMS LOG parameter, even if those files are not being processed under IAM RLS. <strong>The default value is that this option is DISABLED</strong>.</td>
<td></td>
</tr>
<tr>
<td>SAFRC4 – Causes an IAMW28 message to appear when the security subsystem call returns a return code of 4. This message can warn users that their security software is not functioning on IAM files. <strong>Default is Disabled</strong>.</td>
<td></td>
</tr>
<tr>
<td>TURBO – A TURBO mode for Real Time Tuning that will allow IAM to quickly increase the number of buffers it is using when there is heavy physical I/O being performed in short time periods. <strong>The default is TURBO mode is enabled</strong>.</td>
<td></td>
</tr>
<tr>
<td>VAM – Enables IAM's CA-ALLOCATE (formerly SAMS / VAM) support. <strong>The default is Disabled</strong>.</td>
<td></td>
</tr>
<tr>
<td>XESDS - This option will enable 8-byte RBA's as the default for ESDS file under IAM. If disabled then 4-byte RBA's will be used unless overridden with the XESDS CREATE override. <strong>The default is DISABLED</strong>.</td>
<td></td>
</tr>
<tr>
<td>ESDSINTEGRATED</td>
<td>Sets the INTEGRATED OVERFLOW percent for ESDS files. Reserves space in each block for record length changes. Recommended to be set to nonzero if you are using IAM ESDS files that are being updated. <strong>The default is 0</strong>.</td>
</tr>
<tr>
<td>IAMW22LIM</td>
<td>Specifies the amount of storage required for the overflow area index in megabytes that will trigger the IAMW22 reorg recommended message. Valid values are from 16 to 1024. <strong>The default value as IAM is shipped is set to 64 megabytes</strong>. In releases prior to Version 9.2, this was a fixed limit of 16 megabytes, that was too low for many users.</td>
</tr>
</tbody>
</table>
### IMMEDWRITE=

Specifies in which environments IAM is to immediately write out updated data blocks when opened for update processing. Multiple non-conflicting values can be specified. Valid values are:

- **NONE** – Immediate writes are triggered based on the Share Options the type of I/O if running under CICS.
- **ALL** – IAM will always perform immediate writes of any updated data block regardless of environment.
- **CICS** – IAM will immediately write out all updated blocks when running under CICS. The default for CICS is that only randomly updated data blocks are immediately written out to disk.
- **RLS** – IAM will immediately write out any updated data block when the dataset is being processed by IAM/RLS or IAM/PLEX.
- **BATCH** – IAM will immediately write out any updated data block when the dataset is being processed under a batch job step or a TSO user.

The default is **NONE**.

### INDEXSPACE=

Specifies the situations when IAM is to default to using a Data Space to hold the prime and overflow index for Enhanced format files. If IAM is using the Index Space, the original size used is the value set for the DATASPACE value in the Global Options Table, and a maximum size of four (4) times that value, up to 2048, will be set. Valid values are:

- **64BIT** – Indicates that 64 bit storage will be used if available for the index storage. Depending on installation defaults a MEMLIMIT parameter may be needed on the EXEC statement in any job step intending to use 64 bit storage. If 64 bit storage cannot be obtained a DATASPACE will be created to be used for the INDEXSPACE storage. If 64BIT is set it will pertain to all jobs and CICS regions.
- **ALL** - All types of jobs that are accessing Enhanced Format files will use the Index Space.
- **CICS** - Only CICS regions will automatically use an Index Space.
- **NO** - Index Spaces will not be used automatically.

The default is **64BIT**.

### LIMITKEYS

Specifies the number of keys taken in a set when creating an IAM file with a compressed index. You may specify any number from 3 to 64, inclusive.

The default is **32**.
<table>
<thead>
<tr>
<th><strong>Operand</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
</table>
| LOADABWO        | Specifies that IAMSTART (VIFSTART) will load the IAM interfaces for the DFP BWO and RLS Callable catalog information services. If you use a CICS VSAM recovery package that issues these calls, or are running with Transaction Server 1.2 or above, this option must be set to YES. Valid values are:  
YES - Enable the DFP Callable Services interfaces.  
NO - Do not enable the interfaces.  
The default is YES.                                                                                                    |
| LONGKEYS=       | Specifies the minimum key length required to automatically use the long key compression algorithm for Index Compression. Files that use the long key compression that are used with versions of IAM prior to V8.0/12P will use the uncompressed index. Valid values are from 8 to 255. Specifying a value of 64, or perhaps even 32 may improve index compression resulting in less virtual storage for the prime index for large files.  
The default value is 33.                                                                                                    |
| MAXBUFNO        | Specifies the default maximum number of buffers IAM is permitted to acquire during file processing. IAM will use the higher of either this value, or the value for BUFSP. IAM's Real Time Tuning will dynamically adjust the number of buffers used for the file as demand warrants up to this maximum. You may specify a value from 1 to 8192 buffers. (Note that the maximum that will be used for Compatible Format files is 32.)  
The default is 5.                                                                                                           |
| MAXOVERFLOW     | For Compatible Format files, sets the maximum amount of overflow that will be allocated based on the CA% free space value provided in the IDCAMS define.                                                                 |
| MAXREGION       | Specifies the default maximum value, in megabytes, that IAM will dynamically adjust the above the line Region value to. To disable the feature set value to 0. If your CICS regions are already at or above the 512 megabyte size, then increase this value.  
Default is 512.                                                                                                               |
| MAXSECONDARY=   | Default multiplication factors for IAM Dynamic Secondary Space Adjustment feature. The first value is for file loads and the second value is for file updates. To disable this feature, set both values to 0. The Dynamic Secondary Space Adjustment feature is not used for IAM data sets that are DFSMS Extended Format.  
Default values are (10,5).                                                                                                    |
| MINCOMPRESS     | Specifies the minimum acceptable percentage of storage reduction achieved when creating an IAM file to determine if the file qualifies for a compressed index. May be any number from 8 to 40, inclusive.  
The default is 10.                                                                                                           |
<table>
<thead>
<tr>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MULTIVOLUME=</td>
<td>Specifies which space allocation value IAM will use when it appears that a data set will take the next extent on the next volume for Enhanced format IAM files that are not DFSMS Extended Format. Valid values are:</td>
</tr>
<tr>
<td>PRIMARY</td>
<td>- Use the original primary allocation value when a volume switch is anticipated.</td>
</tr>
<tr>
<td>SECONDARY</td>
<td>- Use the original secondary allocation value when a volume switch is anticipated.</td>
</tr>
<tr>
<td>Default is PRIMARY.</td>
<td></td>
</tr>
<tr>
<td>OCOREO%</td>
<td>For Compatible format files only, specifies the amount of virtual storage for expansion of the Overflow index, as a percent of the total capacity of Independent Overflow, to be acquired when an IAM file is OPENed for update processing. 'nnn' may be any number from 1 to 100, inclusive. The default is 10.</td>
</tr>
<tr>
<td>OCOREX%</td>
<td>For Compatible format files only, specifies the amount of virtual storage for expansion of the Overflow index, as a percent of the total capacity of Independent Overflow, to be acquired when more memory is required. 'nnn' may be any number from 1 to 100, inclusive. The default is 10.</td>
</tr>
<tr>
<td>PE</td>
<td>For Compatible format files, specifies the number of blocks of Prime Extension area to be reserved when creating an IAM file. 'nnnnn' may be any number from 0 to 32767, inclusive. The default is 3.</td>
</tr>
<tr>
<td>RECFM</td>
<td>For Compatible format, non-data compressed files, specifies the internal record format IAM is to use for VSAM defines with equal average and maximum record lengths.</td>
</tr>
<tr>
<td>F</td>
<td>- Define the file as fixed.</td>
</tr>
<tr>
<td>V</td>
<td>- Define the file as variable.</td>
</tr>
<tr>
<td>The default is F, however with the default of a file format of enhanced, the RECFM will always be V, unless the CREATE override of FIXED and COMPATIBLE is specified.</td>
<td></td>
</tr>
<tr>
<td>RECTYPE</td>
<td>Specifies the SMF 'user' record type to be written if SMF recording is requested for IAM files, nnn may be a number from 128 to 255, inclusive.</td>
</tr>
<tr>
<td>There is no default value. This is a required field for SMF recording to be requested. Member IAMUSMF in the IAM Installation Control Library is a DSECT of the IAM SMF user record format. The suggested value is 201.</td>
<td></td>
</tr>
<tr>
<td>RELEASE</td>
<td>Specifies the default value for automatic release. Valid values are YES meaning unused disk space in an IAM file is to be released if Secondary allocation value is specified or NO meaning unused disk space is not to be released. The default is YES.</td>
</tr>
</tbody>
</table>
**Operand** | **Description**  
--- | ---  
**RLS=** | Specifies the criteria IAM is to use to automatically determine which IAM files are eligible for IAM RLS processing. There is a combination of two forms of criteria available, which are by Share Options, and/or by data set name. Valid values are:  

**AND** - Indicates that to be automatically eligible for IAMRLS, a data set must have the specified share options, must not be in the data set name exclude list, and must be in the data set name select list.  

**NONE** - Indicates that no data sets are to be considered for automatic eligibility for IAMRLS. Set this option if you are not going to be activating the IAMRLS address space, or if you want to manually direct activity to IAMRLS through the IAM Overrides. This is the default option as IAM is shipped.  

**OPTIONAL** – Allows data sets to be opened without IAM RLS when the IAMRLS address space is not active. This is useful for when the TABLE option has been specified and IAMRLS may not be active on all LPAR’s because IAMRLS has to at least have been active at some point in time for the data set include and exclude tables to be loaded.  

**REQUIRED** – Indicates that IAMRLS MUST BE ACTIVE if a data set is eligible for IAMRLS processing, or when the TABLE Global Option is specified. If IAMRLS is not active, then any attempt to open a potentially IAMRLS eligible data set will be failed. REQUIRED is the default unless the RLS Global Option is set to NONE.  

**OR** - Indicates that to be automatically eligible for IAMRLS, a data set must have either the specified share options or be in the data set name select list. If a data set meets the share option eligibility, then neither the data set name include or exclude lists will be examined. If a data set does not meet the share option eligibility criteria, then it must not be in the exclude list, and must be in the include list. OR is the default if both TABLE and a SHAREx value are specified.  

**SHARE1** - Indicates that data sets with any cross-region share option value (1, 2, 3, or 4) will be eligible for IAMRLS processing.  

**SHARE2** - Indicates that data sets with cross-region share options of 2, 3, or 4 will be eligible for IAMRLS processing.  

**SHARE3** - Indicates that data sets with cross-region share options of 3 or 4 will be eligible for IAMRLS processing.  

**SHARE4** - Indicates that only data sets with a cross-region share option of 4 will be eligible for IAMRLS processing.  

**TABLE** - Indicates that IAM is to search the data set name include and exclude table to determine eligibility for IAMRLS processing, subject to other criteria. The default is that the data set name tables will not take part in eligibility selection for IAMRLS.  

For example, you could code: RLS=(SHARE3,AND,TABLE) which means files that are defined with share option 3 or 4 and are in the include data set name table will be automatically eligible for IAM RLS processing.  

Default values is **NONE**, indicating that no data sets are automatically eligible.
<table>
<thead>
<tr>
<th><strong>Operand</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RLSID</strong></td>
<td>Specifies the 4 character RLSID that will be used as a default when an IAMRLS address space starts if no RLSID is provided in the IAMRLS startup parameters. It will also be the default RLSID used to connect IAM files to an IAMRLS address space if the file was not defined with one or overridden at access time. Any 4 characters may be used. <strong>The default is RLSID=RLS1</strong></td>
</tr>
<tr>
<td><strong>ROUTECODE</strong></td>
<td>Specifies the route code(s) to be used when issuing WTO of WTOR messages. Any number from 1 to 16, inclusive, may be specified. Multiple route codes can be entered as ROUTECODE=(nn,...,nn). <strong>The default is 11 (X'0020').</strong></td>
</tr>
<tr>
<td><strong>SMF</strong></td>
<td>Specifies whether IAM is to write an IAM SMF user record when an IAM file is closed. The IAM SMF user record will contain all of the same information displayed in an IAM INFO Run Time Statistics Report. Member IAMUSMF in the IAM Installation Control Library is a DSECT of the IAM SMF user record's format. <strong>YES</strong> – If a RECTYPE value is specified in IAM's GLOBAL OPTION TABLE an IAM SMF user record will be written whenever an IAM file is closed. <strong>NO</strong> – IAM will not write IAM SMF user records. <strong>The default is NO.</strong></td>
</tr>
<tr>
<td><strong>SORTC</strong></td>
<td>Specifies the amount of storage 'SORT' is to use for the IAM utility programs IAMRECVR and IAMSMFVS. You may specify any number from 10000 to 8000000, inclusive. <strong>The default is 100000.</strong></td>
</tr>
<tr>
<td><strong>SORTM</strong></td>
<td>Specifies the message option to be used by the program 'SORT' if external sorting is required by IAMRECVR or IAMSMFVS. <strong>AC</strong> – all messages to the console <strong>AP</strong> – all messages to the printer (SYSOUT) <strong>CC</strong> – critical messages to the console <strong>CP</strong> – critical messages to the printer <strong>NO</strong> – no messages to be produced <strong>PC</strong> – critical messages to both console and printer <strong>The default is CC.</strong></td>
</tr>
<tr>
<td><strong>SORTPFX</strong></td>
<td>Specifies the ddname prefix to be used by 'SORT' invoked by IAMRECVR and IAMSMFVS. If the string specified is less than 4 characters, a dollar sign ($) fill character will be used. <strong>The default is SORT.</strong></td>
</tr>
<tr>
<td><strong>STORCLASS</strong></td>
<td>Specifies the SMS storage class to be used if SMS is active and a storage class was not specified on the define request, or the installation ACS routines did not assign an SMS storage class. If this option is blanks, no SMS storage group will be assigned. To reset this option to blanks, specify STORCLASS=' '. <strong>The default is blanks.</strong></td>
</tr>
</tbody>
</table>
### IAMZAPOP – ZAP Command

<table>
<thead>
<tr>
<th><strong>Operand</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VAROVERFLOW</strong></td>
<td>Specifies whether IAM is to use variable length overflow for IAM files when they are defined.</td>
</tr>
<tr>
<td></td>
<td>Valid values are:</td>
</tr>
<tr>
<td></td>
<td><strong>YES</strong> - Set files as being eligible for variable length overflow when they are defined.</td>
</tr>
<tr>
<td></td>
<td><strong>NO</strong> - Do not set files as eligible for variable length overflow when they are defined.</td>
</tr>
<tr>
<td></td>
<td><strong>Default value is YES.</strong></td>
</tr>
<tr>
<td><strong>VSAMBLOCKF</strong></td>
<td>Specifies the default blocking factor (number of blocks per track) to be used when an IDCAMS DEFINE does not specify CISIZE. Any value from 1 to 15 inclusive may be specified.</td>
</tr>
<tr>
<td></td>
<td><strong>The default value is 4</strong> (four blocks per track).</td>
</tr>
<tr>
<td><strong>VSAMWTO</strong></td>
<td>For Compatible format files only, specifies the action to be taken under the VSAM interface, when an IAM file is not available.</td>
</tr>
<tr>
<td></td>
<td><strong>YES</strong> – An IAMW02 message is generated and the operator must reply Retry, Wait, or Cancel.</td>
</tr>
<tr>
<td></td>
<td><strong>NO</strong> – The Open is failed.</td>
</tr>
<tr>
<td></td>
<td><strong>The default is NO.</strong></td>
</tr>
<tr>
<td><strong>WORKDDNAME</strong></td>
<td>Specifies the ddname of the work file used during an IAM load function. The work file will be used if IAM is not using a Data Space to hold the index.</td>
</tr>
<tr>
<td></td>
<td>The default is IAMWKDD if it is in the JCL, otherwise the file will be dynamically allocated.</td>
</tr>
<tr>
<td><strong>WORKPRIMARY</strong></td>
<td>Specifies the primary allocation in tracks of the work file dynamically allocated during an IAM load function. Any value from 1 to 65535, inclusive, may be specified.</td>
</tr>
<tr>
<td></td>
<td><strong>The default is 30</strong> (tracks).</td>
</tr>
<tr>
<td><strong>WORKSECONDARY</strong></td>
<td>Specifies the secondary allocation in tracks for the work file dynamically allocated during an IAM load function. Any value from 1 to 65535, inclusive, may be specified.</td>
</tr>
<tr>
<td></td>
<td><strong>The default is 30</strong> (tracks).</td>
</tr>
<tr>
<td><strong>WORKUNIT</strong></td>
<td>Specifies the unit name to be used when dynamically allocating the work file used during an IAM load function and during a DEFINE of an IAM data set with the IAM ANYVOL support.</td>
</tr>
<tr>
<td></td>
<td><strong>The default is SYSDA.</strong></td>
</tr>
</tbody>
</table>
91.04 IAMZAPOP JCL Examples

The following examples illustrate some of the ways of executing the GLOBAL OPTION CHANGE FACILITY.

Example 1
The user wishes to display the present Global values.

```jcl
//PRINT EXEC PGM=IAMZAPOP
//SYSPRINT DD SYSOUT=A
//SYSLIB DD DSN=iam.library,DISP=SHR
//SYSPRINT DD *
PRINT
/*
```

Figure 91-4: Example of Printing out the IAM Global Options (EX9104A)

Example 2
The user enables Automatic Data Compression for files that are 150 tracks or larger. The ZAP command will modify the IAM Option Table.

```jcl
//ZAP EXEC PGM=IAMZAPOP
//SYSPRINT DD SYSOUT=A
//SYSLIB DD DSN=iam.library,DISP=SHR
//SYSPRINT DD *
ZAP DATACOMPRESS=150
/*
```

Figure 91-5: Example of Using IAMZAPOP to set Default for Data Compression (EX9104B)

Example 3
Reset all of the Global Option values to their original values supplied on installation tape.

```jcl
//RESET EXEC PGM=IAMZAPOP
//SYSPRINT DD SYSOUT=A
//SYSLIB DD DSN=iam.library,DISP=SHR
//SYSPRINT DD *
RESET
/*
```

Figure 91-6: Example of Resetting IAM Global Options to Initial Values (EX9104C)
Example 4
Enable the IAM SMF recording option so an SMF user record is written every time an IAM file is CLOSEd. IAM's SMF user record type is to be type 201.

```plaintext
//SETSMF EXEC PGM=IAMZAPOP
//SYSPRINT DD SYSOUT=A
//SYSLIB DD DSN=iam.library,DISP=SHR
//SYSIN DD *
   ZAP RECTYPE=201,SMF=YES
/*
```

Figure 91-7: Example of Using IAMZAPOP to Enable IAM SMF Recording (EX9104D)

Example 5
COPY all of the changed Global Option values from an IAM V9.2 load library to another IAM V9.2 load library, or to a library that will be used with IAM V9.2.

Do not use this to copy the options from a prior version of IAM.

```plaintext
//COPY EXEC PGM=IAMZAPOP
//SYSPRINT DD SYSOUT=A
//SYSLIB DD DSN=iam.library,DISP=SHR
//COPYTODD DD DSN=iam.new.library,DISP=SHR
//SYSIN DD *
   COPY
/*
```

Figure 91-8: Example of Using IAMZAPOP to Copy a Global Options Table (EX9104E)

Example 6
Using the AUDIT command to list any Global Option values that have been changed from their distributed defaults.

```plaintext
//AUDIT EXEC PGM=IAMZAPOP
//SYSPRINT DD SYSOUT=A
//SYSLIB DD DSN=iam.library,DISP=SHR
//SYSIN DD *
   AUDIT
/*
```

Figure 91-9: Example of Using IAMZAPOP AUDIT to Determine Which Global Option Values Have Been Changed (EX9104F)
Example 7 In this example, the RLS override is changed to automatically select those data sets defined with Share Option 3 or 4, or that are included in a data set name table, that is provided to IAM RLS. Use of IAMRLS is set to optional, which means data sets can be opened without IAMRLS being active.

```
//SETRLS EXEC PGM=IAMZAPOP
//SYSPRINT DD SYSOUT=A
//SYSLIB DD DSN=iam.library,DISP=SHR
//SYSIN DD *
    ZAP RLS=(SHARE3,OR,TABLE,OPTIONAL)
/*
```

Figure 91-10: Example of Using IAMZAPOP to change the RLS automatic selection criteria (EX9104G)
Reader’s Comment Form
IAM User Documentation V9.2
INNOVATION DATA PROCESSING

If you have comments on this manual, including:
Errors in the text or typographical errors
☐ Clarity
☐ Suggestions for improvement in the manual
☐ Suggestions for improvement in the product
☐ Any other comments

Please complete this form and fax it to INNOVATION DATA PROCESSING at 973-890-7147 (in Europe you may fax it to your local INNOVATION office as shown on the front page of the manual). You may also e-mail your comments to INNOVATION at support@fdrinnovation.com (be sure to identify the manual name in the message).

Your name: _____________________________________________
Company name: ___________________________________________
Mailing address: __________________________________________
E-mail address: ___________________________________________