User’s Guide

Version 2 Release 2
Contents

Figures ............................................................... vii
Tables ................................................................. ix

About this book .................................................. xi
Who should read this book ........................................ xi
Conventions used in this book .................................... xi
Terminology used in this book ................................... xii
Service updates and support information ..................... xii
Using LookAt to look up message explanations .............. xii
Where to find information ....................................... xiii
Accessibility features ............................................. xiii
How to send your comments .................................. xiii

Summary of Changes .............................................. xv

Chapter 1. Introduction to Archive Log Accelerator .......... 1
  What does Archive Log Accelerator do? ....................... 1
    Features ..................................................... 2
  Benefits of using Archive Log Accelerator .................. 2
  Archive Log Accelerator environment ......................... 4
    Overview .................................................. 4
    Implementing DFSMS compression and data striping .......... 5
    Supported data set types ................................ 6
    Using Archive Log Accelerator in a data sharing environment 6
    Using Archive Log Accelerator with other DB2 tools and utilities 6
  Archive Log Accelerator components ......................... 7
    Started task ............................................... 7
    Control file .............................................. 7
    Log file ................................................. 7
    Compression optimizer .................................... 7
    Hardware compression modeling tool ....................... 7
    ISPF interface ........................................... 8
  Uses for Archive Log Accelerator ............................. 8
    Accelerating archive log offload processing ............... 8
    Accelerating DB2 data recovery .......................... 8
    Conserving storage space and reducing storage costs ...... 8
  Security considerations ..................................... 8
  Performance considerations .................................. 9
    Archive log placement ................................... 9
    Placing archive logs on tape ............................. 9
  Prerequisites ................................................ 10
    Hardware requirements .................................. 10
    Software requirements .................................. 10

Chapter 2. Configuring Archive Log Accelerator .......... 11
  Configuration summary ...................................... 11
  Step 1: Check authorization requirements .................. 12
  Step 2: Create the DB2PARMS VSAM control file ............ 12
  Step 3: Create the log file ................................ 12
  Step 4: Configure and initiate the started task ............ 13
    Modify the started task JCL ............................. 13
    Initiate the started task ............................... 13
# Figures

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DB2 archive log offload processing (&quot;slow offload problem&quot;)</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Conserve storage space by optimizing DFSMS compressed archive logs</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Architec Log Accelerator environment</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>DB2 Archive Log Accelerator for z/OS main menu</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>Enter DB2 System Parameters screen</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>Update Parameters for DB2 Subsystem screen</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>Archive Log Accelerator Configuration screen</td>
<td>19</td>
</tr>
<tr>
<td>8</td>
<td>Overview of Archive Log Accelerator processing</td>
<td>27</td>
</tr>
<tr>
<td>9</td>
<td>Overview of extended format archive log processing</td>
<td>28</td>
</tr>
<tr>
<td>10</td>
<td>Extended format archive log processing flow</td>
<td>29</td>
</tr>
<tr>
<td>11</td>
<td>Archive Log Accelerator optimizing a compressed archive log</td>
<td>30</td>
</tr>
<tr>
<td>12</td>
<td>Non-extended format archive log processing flow</td>
<td>31</td>
</tr>
<tr>
<td>13</td>
<td>Example archive log parameter settings for striped logs (accelerating archive log offload processing)</td>
<td>34</td>
</tr>
<tr>
<td>14</td>
<td>Example archive log parameter settings for DFSMS compressed logs (accelerating DB2 data recovery)</td>
<td>36</td>
</tr>
<tr>
<td>15</td>
<td>Example archive log parameter settings for optimized archive logs (optimizing DFSMS compression)</td>
<td>38</td>
</tr>
<tr>
<td>16</td>
<td>Current Status screen</td>
<td>51</td>
</tr>
<tr>
<td>17</td>
<td>Archive Log Summary screen</td>
<td>55</td>
</tr>
<tr>
<td>18</td>
<td>Compression History screen</td>
<td>56</td>
</tr>
<tr>
<td>19</td>
<td>BSDS Archive History screen</td>
<td>57</td>
</tr>
<tr>
<td>20</td>
<td>Using SMS contructs and ACS routines to enable DFSMS compression for archive logs</td>
<td>59</td>
</tr>
<tr>
<td>21</td>
<td>Example data class (defined for compression)</td>
<td>60</td>
</tr>
<tr>
<td>22</td>
<td>Sample DATACLAS ACS routine (DFSMS compression)</td>
<td>61</td>
</tr>
<tr>
<td>23</td>
<td>Sample storage class defined for data striping</td>
<td>62</td>
</tr>
<tr>
<td>24</td>
<td>Example DATACLAS ACS routine (data striping)</td>
<td>63</td>
</tr>
<tr>
<td>25</td>
<td>ALCMODEL member contents</td>
<td>65</td>
</tr>
<tr>
<td>26</td>
<td>Sample hardware compression modeling tool report</td>
<td>67</td>
</tr>
</tbody>
</table>
Tables

1. Placing archive logs on disk or tape .................................................. 9
2. Steps to configure Archive Log Accelerator ........................................ 11
3. Log file JCL modifications ................................................................. 12
4. Started task JCL modifications .......................................................... 13
5. ALC member JCL modifications .......................................................... 14
6. ALCCLIST member JCL modifications .............................................. 15
7. Optional started task parameters ....................................................... 23
8. Examples of work data set name prefixes for optimization ....................... 31
9. Determining archive log parameter settings based on archive log management goals .......................................................... 32
10. Copy1 and Copy 2 archive log parameter settings and corresponding results .......................................................... 39
11. Migration information within this user guide ...................................... 70
12. Message severity codes .................................................................. 71
About this book

This book provides instructions for configuring and using DB2® Archive Log Accelerator for z/OS®. The book contains detailed information, such as field descriptions, sample JCL, and line commands, to help DB2 Archive Log Accelerator users fully benefit from the use of this product.

This book is designed to help database administrators, system programmers, and DB2 system administrators perform these tasks:

- Configure DB2 Archive Log Accelerator
- Use DB2 Archive Log Accelerator to compress, copy, and decompress DB2 archive logs

The technical changes for this edition are summarized under “Summary of Changes” on page xv. Specific changes since the previous edition of the book are indicated by a vertical bar (|) to the left of the change. Editorial changes that have no technical significance are not noted.

Always check the DB2 and IMS™ Tools Library page for the most current version of this publication:

www.ibm.com/software/data/db2imstools/library.html

Who should read this book

This book is intended for those persons responsible for installing, configuring, and using DB2 Archive Log Accelerator and assumes a working knowledge of:

- The OS/390® or z/OS operating system
- DB2
- ISPF
- SMP/E

Conventions used in this book

This book uses the following highlighting conventions:

- **Boldface type** indicates commands or user interface controls such as names of fields, folder, icons, or menu choices.
- **Monospace type** indicates examples of text that you enter exactly as shown.
- **Italic type** indicates variables that you should replace with a value. It is also used to indicate book titles and to emphasize significant words.

The following labels identify significant elements within this book:

- **Definition**: is used to identify and define terminology unique to this product.
- **Example**: is used to identify example code or scenarios.
- **In this chapter**: is used to identify the significant subsections within each chapter.
- **Recommendation**: is used to provide guidance when more than one option is available.
- **Related reading**: is used to refer you to other publications that contain relevant information.
- **Requirement**: identifies a condition that must be met to ensure that the product is functional.
- **Restriction**: is used to identify a restriction or limitation with this product or an associated procedure.

**Terminology used in this book**

In this book, DB2 Archive Log Accelerator for z/OS is referred to as “DB2 Archive Log Accelerator.” In cases where the context makes the meaning clear, DB2 Archive Log Accelerator is referred to as “Archive Log Accelerator.”

**Service updates and support information**

To find service updates and support information, including software FixPaks, PTFs, Frequently Asked Question (FAQs), technical notes, troubleshooting information, and downloads, refer to the following Web page:

[www.ibm.com/software/data/db2imstools/support.html](http://www.ibm.com/software/data/db2imstools/support.html)

**Using LookAt to look up message explanations**

LookAt is an online facility that lets you look up explanations for most messages that you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from the following locations to find IBM® message explanations for z/OS elements and features, z/VM®, and VSE:

- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations, using LookAt from a TSO/E command line (for example, TSO/E prompt, ISPF, or z/OS UNIX® System Services running OMVS).
- Your Windows® workstation. You can install code to access IBM message explanations on the z/OS Collection kit (SK3T-4269), using LookAt from a Windows DOS command line.
- Your wireless handheld device. You can use the LookAt Mobile Edition with a handheld device that has a wireless access and an Internet browser (for example, Internet Explorer for Pocket PCs, Blazer, or Eudora for Palm OS, or Opera for Linux™ handheld devices). Link to the LookAt Mobile Edition from the LookAt Web site.

You can obtain code to install LookAt on your host system or Windows workstation from a disk on your z/OS Collection kit (SK3T-4269), or from the LookAt Web site (click Download, and select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files that are available during the download process.
Where to find information

The DB2 and IMS Tools Library Web page provides current product documentation that you can view, print, and download. To locate publications with the most up-to-date information, refer to the following Web page:

[www.ibm.com/software/data/db2imstools/library.html](http://www.ibm.com/software/data/db2imstools/library.html)

IBM Redbooks™ that cover DB2 and IMS Tools are available from the following Web page:

[www.ibm.com/software/data/db2imstools/support.html](http://www.ibm.com/software/data/db2imstools/support.html)

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully. The major accessibility features in DB2 Archive Log Accelerator enable users to:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
  - z/OS ISPF User’s Guide, Volume 1, SC34-4822
  - z/OS TSO/E Primer, SA22-7787
  - z/OS TSO/E User’s Guide, SA22-7794

These guides describe how to use ISPF, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other DB2 Archive Log Accelerator documentation:

- Use the online reader comment form located at:

- Send your comments by e-mail to dmtinfo@us.ibm.com. Be sure to include the name of the book, the part number of the book, the version of DB2 Archive Log Accelerator, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

- Print and fill out the reader comment form located at the back of this book. You can either give the completed form to your local IBM branch office or IBM representative, or you can send it to the address printed on the reader comment form.
Summary of Changes

Significant improvements and enhancements have been made to DB2 Archive Log Accelerator for V2.2. This section describes the product modifications and refers you to relevant sections of this book for more information:

SC18-7405-02 — December 2004

- For an overview of Archive Log Accelerator for V2.2, including updated environment, component, and hardware and software requirement information, see Chapter 1, “Introduction to Archive Log Accelerator,” on page 1.
- The compression modes (Passthru and Cache) have been eliminated. As a result, the Compress Mode option has been removed from the Archive Log Accelerator Configuration screen and the CompMode column has been removed from the Current Status screen. See "Step 7: Configure Archive Log Parameters" on page 19 and "Accessing the Current Status screen" on page 51 for details.
- The Archive Log Accelerator Configuration screen has been redesigned. In addition to removing the Compress Mode option, the Copy 1a and Copy 2a fields (including the corresponding Unit, Space, and Dataset Name Prefix fields) have been removed, and an Optimize option for DFSMS compressed archive logs has been added. See “Step 7: Configure Archive Log Parameters” on page 19 for details including updated field descriptions.
- Archive Log Accelerator processing has been modified to reflect the changes to the product for version 2.2. See “Understanding how Archive Log Accelerator processes archive logs” on page 27 for updated information on how Archive Log Accelerator processes archive logs.
- Archive log scenarios have been updated to reflect product modifications. See Chapter 4, “Archive Log Scenarios,” on page 43 for details.
- Information that describes how to modify SMS constructs and ACS routines to implement DFSMS compression and data striping for archive logs has been added. See Appendix A, “Modifying SMS Constructs and ACS Routines,” on page 59 for more information.
- The batch processing feature has been replaced with a hardware compression modeling tool. As a result, the ALBCMP and ALDCMP members have been removed from the sample library, and JCL for the modeling tool has been added. For more information about the hardware compression modeling tool, see Appendix B, “Using the Hardware Compression Modeling Tool,” on page 65.
- An option that enables you to delete archive log parameters for a specific DB2 subsystem has been added to the Enter DB2 System Parameters screen. See “Deleting archive log parameters settings” on page 21 for details.
- The Purge (P) command has been removed from the BSDS Archive History screen. See "Viewing BSDS archive history" on page 57 for more information.

SC18-7405-01 — February 2004

- Archive Log Accelerator environment, component, and hardware and software requirement information has been added. See Chapter 1, “Introduction to Archive Log Accelerator,” on page 1 for details.
- Additional information about Archive Log Accelerator compression modes has been added.
- Information has been added for Archive Log Compression Tool Version 1.1 users to transition to DFSMS compression.
• The configuration summary table has been updated to include job names. For more information, see “Configuration summary” on page 11 for details.

• The Optimize parameters and the Copy 1a and Copy 2a Catalog parameters have been removed from the Archive Log Configuration screen. For more information, see “Step 7: Configure Archive Log Parameters” on page 19.

• The Copy1 and Copy 2 Catalog parameter description has been modified, for details, see “Step 7: Configure Archive Log Parameters” on page 19.

• Additional information has been provided to help you determine your configuration settings, see “Determining archive log parameter settings” on page 32 for details.

• Additional archive log scenarios have been added, including a scenario that describes how to convert an Archive Log Compression Tool Version 1.1 compressed log to a DFSMS compressed log.

• A bibliography has been added. See “Bibliography” on page 85.

SC18-7405-00 — June 2003

• The product name and version has been updated (from DB2 Archive Log Compression Tool for z/OS V1.1 to DB2 Archive Log Accelerator for z/OS V2.1).

• A new chapter, to describe how DB2 Archive Log Accelerator supports extended format archive log data sets, has been added.

• A new chapter, to describe DB2 Archive Log Accelerator processing, has been added.

• Information has been added to describe data striping and DFSMS compression support has been added where applicable.

• Information has been added to describe DSN1LOGP support.

• Information has been added to describe new messages.

• A glossary has been added. See “Glossary” on page 83.
Chapter 1. Introduction to Archive Log Accelerator

This chapter introduces you to the functionality provided by Archive Log Accelerator (ALC), including its features and key benefits. This chapter also describes the product components and environment as well as the prerequisite hardware and software that must be installed before using Archive Log Accelerator.

In this chapter:

- What does Archive Log Accelerator do?
- "Archive Log Accelerator environment" on page 4
- "Archive Log Accelerator components" on page 7
- "Uses for Archive Log Accelerator" on page 8
- "Security considerations" on page 8
- "Performance considerations" on page 9
- "Prerequisites" on page 10

What does Archive Log Accelerator do?

Maintaining archive logs is an essential part of data management. Archive logs provide the protection required to recover from all types of disasters, both data and hardware driven. However, archive logs can create storage problems because they require a considerable amount of space. While it is possible to store archive logs on tape, recovering the logs can be time consuming, especially when waiting for devices or volumes to become available.

Archive Log Accelerator helps alleviate these problems, and reduce the overhead associated with database log management to balance the increases in archive log growth, by enabling you to:

- Stripe archive logs to accelerate archive log offload processing
- DFSMS compress archive logs to accelerate DB2 data recovery
- Optimize DFSMS compressed archive logs to further reduce the size of logs and conserve valuable storage space

Prior to Archive Log Accelerator, you could not take advantage of the benefits of DFSMS compression and data striping for archive logs. This is because DB2 uses BDAM (Basic Direct Access Method) to read archive log data sets from DASD and DFSMS/MVS cannot read extended format data sets, the data set format required for DFSMS compression and data striping, using BDAM.

Archive Log Accelerator removes this restriction for DB2 archive logs by automatically detecting whether or not the archive log data set is an extended format data set and, if so, managing the unsupported BDAM access on behalf of DB2. As a result, archive logs can be defined as extended format data sets and you can take advantage of DFSMS compression and data striping benefits for archive logs.

Note: Archive Log Accelerator does not stripe or DFSMS compress archive logs—instead, it enables DB2 to read compressed and striped archive logs to allow the logs to be compressed or striped. Thus, it is not possible to use DFSMS to compress or stripe archive log data sets without the use of Archive Log Accelerator.
Features

Archive Log Accelerator provides three types of "accelerator" options to choose from. You can take advantage of any combination of these accelerators to best suit your needs.

- **DFSMS Striping Accelerator**—This accelerator allows you to take advantage of high performance striping I/O for all read and write activity against the archive logs. Striping your archive logs enables you to accelerate the archive log offload process.

  To take advantage of this accelerator, your storage administrator must use SMS to define your archive logs as extended format *striped* data sets.

- **DFSMS Hardware Compression Accelerator**—This accelerator allows you to take advantage of hardware compression when archive logs are written (compressed logs are automatically decompressed when they are read). Using DFSMS to compress your archive logs enables you to accelerate DB2 data recovery.

  To take advantage of this accelerator, your storage administrator must use SMS to define your archive logs as extended format *compressed* data sets.

- **DFSMS Optimized Hardware Compression Accelerator**—This accelerator provides an additional level of compression above and beyond the DFSMS Compression Accelerator, allowing you to experience improved compression ratios. Optimizing your archive logs enables you to reduce the size of the logs, thus reducing the amount of storage space required to store the archive logs.

  To take advantage of this accelerator, you must configure Archive Log Accelerator with the **Optimize** archive log parameter set to **Y**. When you enable this parameter, Archive Log Accelerator performs an optimization step against a working copy of the archive log to allow for improved compression ratios. (After the working copy of the log is optimized, the original archive log is deleted and the working copy is renamed to the archive log name.

  **Note:** Whether you are compressing or striping archive logs, Archive Log Accelerator must be active during DB2 recovery since it performs the read operations on behalf of DB2. If you define your archive logs as DFSMS compressed or striped data sets and Archive Log Accelerator is not active, DB2 will not be able to read the archive logs for any recovery operations such as undo, redo, and restore.

For more information about DFSMS compression and data striping, see *DFSMS/MVS General Information*, *DFSMS Introduction*, *DFSMS/MVS Implementing System-Managed Storage*, and *DFSMS/MVS Using Data Sets*.

Benefits of using Archive Log Accelerator

The accelerator features provided by Archive Log Accelerator can reduce the overhead associated with database log management to balance the increases in archive log growth as well as maximize data availability and reduce storage costs as described in "Accelerates archive log offload processing" on page 3, "Accelerates DB2 data recovery" on page 3, and "Conserves storage space and reduces storage costs" on page 3.
**Accelerates archive log offload processing**

In installations with high logging rates or short, periodic bursts of high transactions, the archive log offload process can take too long (a situation referred to as the "slow offload problem"). This happens because log records are being written to the active log faster than DB2 can offload the active log. As a result, the active log fills up and starts an archive offload. As this is taking place, the next active log fills up and starts an archive offload, and so on, until all of the active logs are full and the archive offloads are not complete. When this happens, DB2 stops running until it catches up with the archive log offloads, thus, the archive log process is slowed.

![Diagram of DB2 archive log offload processing](image)

*Figure 1. DB2 archive log offload processing ("slow offload problem")*

Using Archive Log Accelerator, you can stripe the archive logs and speed up the offload process so DB2 can keep up with the archive offloads. Striping the archive log data sets improves I/O performance and reduces the archive log window. Improved QSAM buffer management for striped data sets also reduces processing times for striped archive logs since DB2 writes the archive log data sets using QSAM. As a result, the archive logs can be written faster and offload processing is accelerated.

**Accelerates DB2 data recovery**

Archive Log Accelerator speeds DB2 data recovery (specifically, recovery log processing) by providing the ability to use DFSMS to compress your archive logs. When DB2 archive logs are compressed, the number of I/Os needed to retrieve the log records decreases, thus accelerating the recovery processes that read from the logs. As a result, data availability is maximized by leveraging DASD storage that contains compressed archive logs.

**Conserves storage space and reduces storage costs**

To maximize your compression savings, you can choose to optimize DFSMS compressed archive logs. Requesting that Archive Log Accelerator optimize the DFSMS compressed archive logs further reduces the size of logs and reduces the amount of storage space required to retain the archive logs, thereby reducing your storage costs. Archive Log Accelerator optimizes the compression by removing...
unneeded records from the archive log.

Without Archive Log Accelerator

![Diagram showing archive logs without optimization](image)

With Archive Log Accelerator

![Diagram showing archive logs with optimization](image)

By optimizing DFSMS compressed archive logs, you can reduce your storage costs by keeping more archive logs in the same amount of space.

*Figure 2. Conserve storage space by optimizing DFSMS compressed archive logs*

For example, if you have 496 DFSMS compressed archive logs (and each compressed log is 5075 blocks in size) and you optimize the logs so that each log is 4992 blocks in size, you could potentially store 504 optimized, compressed archive logs in the same amount of space as the 496 (uoptimized) compressed archive logs.

*Note:* Your compression savings may vary. To determine your compression savings, Archive Log Accelerator provides a hardware compression modeling tool that shows compression statistics for optimized and uoptimized DFSMS compressed archive logs. For more information, see “Hardware compression modeling tool” on page 7.

---

**Archive Log Accelerator environment**

This section describes the Archive Log Accelerator environment.

**Overview**

As shown in [Figure 3](image), Archive Log Accelerator runs as a started task. The started task establishes and provides the operating environment required by Archive Log...
Accelerator to perform archive log-related tasks.

When the Archive Log Accelerator started task is initiated, it reads a control file that is created during the installation of Archive Log Accelerator. The control file contains the configuration information that determines where and how Archive Log Accelerator performs processing. When DB2 starts an offload process to archive the log (in response to a full active log or an ARCHIVE LOG command) Archive Log Accelerator handles the archive logs according to the configuration information in the control file.

Implementing DFSMS compression and data striping

Implementing DFSMS compression or data striping for your archive logs is transparent to DB2 and to Archive Log Accelerator (no modifications to DB2 or Archive Log Accelerator are required). To stripe or DFSMS compress archive logs, the archive log data sets must be defined as extended format data sets using DFSMS. Because extended format data sets are SMS-managed, your storage administrator must define and assign the appropriate SMS constructs to allow the
archive logs to be compressed or striped. To do this, the storage administrator adds the archive log data set prefixes (the ARCPFX1 and ARCPFX2 values specified in the DB2 subsystem DSNZPARM configuration) to the appropriate SMS constructs as required by DFSMS. After translating and activating the appropriate ACS routine, no further action is needed. Archive Log Accelerator automatically detects whether or not the archive log is extended format and allows SMS to stripe or compress the archive logs as defined by your storage administrator.

For more information about implementing data striping and DFSMS compression for archive logs, see Appendix A, “Modifying SMS Constructs and ACS Routines,” on page 59.

Supported data set types

Archive Log Accelerator supports two types of data sets:

- SMS-managed extended format data sets defined for DFSMS compression, data striping, or both (DFSMS compression and data striping)
- Non-SMS managed, non-extended format data sets (tape only)

Using Archive Log Accelerator in a data sharing environment

Archive Log Accelerator can be used in a data sharing environment—simply configure Archive Log Accelerator for each member in the data sharing group where you want to use DFSMS to compress or stripe archive logs. For more information about configuring Archive Log Accelerator for a data sharing environment, see “Configuring Archive Log Accelerator for a data sharing environment” on page 24.

Using Archive Log Accelerator with other DB2 tools and utilities

Archive Log Accelerator is an integral part of a portfolio of recovery management tools that includes the DB2 Log Analysis Tool and the DB2 Change Accumulation Tool. Used in combination with Archive Log Accelerator, these tools provide additional options that help improve performance during recovery processes.

DB2 Log Analysis Tool for z/OS

The Log Analysis Tool allows direct analysis of compressed (unoptimized) or non-compressed archive logs, thus saving you time by eliminating the need to decompress logs prior to using analysis, or other Log Analysis Tool functions. For more information about using the Log Analysis Tool, see the DB2 Log Analysis Tool for z/OS User’s Guide.

Change Accumulation Tool

The Change Accumulation Tool can be used to facilitate the recovery process by reducing the amount of log records and archive log files needed for processing. There is no need to uncompress archive logs as the Change Accumulation Tool can read logs that have been compressed using DFSMS. For more information about using the Change Accumulation Tool, see the DB2 Change Accumulation Tool User’s Guide.

DB2 Administration Tool

If your installation uses the DB2 Administration Tool, you can include Archive Log Accelerator in the launch panel. For more information, see “Step 10: (Optional) add Archive Log Accelerator to the DB2 Administration Tool Launchpad” on page 24.

DSN1LOGP

You can read compressed (unoptimized) archive logs using the log print utility DSN1LOGP.
**Archive Log Accelerator components**

Archive Log Accelerator consists of the following components.

**Started task**

Archive Log Accelerator requires a started task component that runs in its own address space. The started task establishes and provides the operating environment required by Archive Log Accelerator. When you use Archive Log Accelerator in a TSO environment, it communicates with the started task to perform necessary tasks required by the various Archive Log Accelerator processes.

*Note:* The Archive Log Accelerator started task must be active to use DFSMS to stripe or compress archive logs. If the started task is not available, all requests to read a compressed or striped archive log can produce undesirable, and potentially erroneous, results.

The started task does not cause activity to occur or undesirable resource use. Once active, it services requests from Archive Log Accelerator users that are operating in a TSO environment. Most of its activity is spent waiting for requests of this nature. If Archive Log Accelerator is not actively operating on archive logs for a target DB2, the started task resource use is minimal. For more information about the started task, see “Step 4: Configure and initiate the started task” on page 13.

**Control file**

The Archive Log Accelerator control file is a VSAM data set that is created during the installation and configuration process. This file contains configuration information specific to Archive Log Accelerator. When you modify the Archive Log Accelerator configuration, the contents of the control file are updated to reflect your configuration changes. For more information, see “Step 2: Create the DB2PARMS VSAM control file” on page 12.

**Log file**

During the installation and configuration process, Archive Log Accelerator generates a log file which records the actions taken by Archive Log Accelerator started task. The record of the product actions can be viewed at any time using the Archive Log Reporting option available from the DB2 Archive Log Accelerator for z/OS main menu. For more information about the log file, see “Step 3: Create the log file” on page 12. For more information about viewing archive log reports, see Chapter 6, “Obtaining Archive Log Reports,” on page 55.

**Compression optimizer**

To maximize compression benefits and optimize storage space, Archive Log Accelerator provides an Optimize option that enables you to further decrease the size of archive logs that have been compressed using DFSMS. When you select the Optimize option on the Archive Log Accelerator Configuration screen, Archive Log Accelerator optimizes the compressed log by removing unneeded records from the archive log. For more information, see “Optimization processing” on page 30 and “Step 7: Configure Archive Log Parameters” on page 19.

**Hardware compression modeling tool**

Archive Log Accelerator provides a hardware compression modeling tool that generates compression statistics that can help you determine how to use Archive Log Accelerator to best suit your specific needs. The tool enables you to evaluate
the benefits of DFSMS compression—with and without compression optimization.
For more information, see Appendix B, “Using the Hardware Compression Modeling Tool,” on page 65.

**ISPf interface**

Archive Log Accelerator provides an ISPF full screen interface that enables you to:

- Configure the DB2 subsystems on which you want Archive Log Accelerator to handle your archive logs.
- View the current status of archive log activity and configuration settings for a specific DB2 to help you determine the level of activity related to your archive logs.
- Access archive log reports including: a summary of archive log information listed by DB2 subsystem—and historical reports of archive logs that have been compressed for a specific DB2, as well as BSDS data for a specific DB2.

**Uses for Archive Log Accelerator**

Archive Log Accelerator enables you to effectively maximize your database recovery efforts and conserve valuable system resources as described in these examples.

**Accelerating archive log offload processing**

If your installation is experiencing high logging rates, you can use Archive Log Accelerator to speed up offload processing. This capability is made possible by striping the archive log data sets across multiple volumes. As a result, the archive logs can be written faster and offload processing is accelerated.

**Accelerating DB2 data recovery**

If you need to keep more of your archive logs on disk instead of tape, you can use Archive Log Accelerator to take advantage of DFSMS hardware compression to compress archive logs. Data availability is maximized by leveraging DASD storage that contains compressed archive logs.

**Conserving storage space and reducing storage costs**

If you need to reduce the amount of space required to hold archive log data, use Archive Log Accelerator to allow your archive logs to take advantage of DFSMS hardware compression. This enables you to reduce amount of space required to hold archive log data, as well as improve buffering and caching, and reducing I/O rates. To maximize the savings of compression, you can request that Archive Log Accelerator optimize the compression by removing unneeded records.

**Note:** The hardware compression modeling tool can help you determine your compression savings. For more information, see Appendix B, “Using the Hardware Compression Modeling Tool,” on page 65.

For specific examples, including sample configurations, see Chapter 3, “Using Archive Log Accelerator,” on page 27 and Chapter 4, “Archive Log Scenarios,” on page 43.

**Security considerations**

Archive Log Accelerator requires no additional security measures outside of standard DB2 security.
Performance considerations

Archive logs can be placed on disk or tape. However, where you place the archive logs affects the recovery process. Review the performance considerations described in this section to determine where you should place your archive logs to best suit your installation’s requirements and goals.

Archive log placement

Table 1 provides an overview of some factors to consider when placing archive logs on disk or tape.

<table>
<thead>
<tr>
<th>Media</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| Disk  | • Supports extended format data sets, allowing the archive logs to be compressed or striped.  
• Facilitates recovery processes by reducing the amount of time it takes to write the archive log.  
• Provides increased data availability due to increased performance. | • Can be more expensive media on which to store archive logs. |
| Tape  | • More cost-effective to store archive logs for disaster recovery or archival purposes. | • Does not support extended format data sets.  
• Archive logs must be sized to fit on one cartridge.  
• Tape compression is only supported for non-extended format archive logs that do not span more than one tape cartridge.  
• The DSNZPARRM parameter (DSN6ARVP COMPACT) allows compaction. However, the data on the tape can only be read using a device that supports improved data recording capability (IDRC). |

Placing archive logs on tape

With Archive Log Accelerator, you can write the files to disk using DFSMS compression, and then direct the files to tape using DFHSM. This method is preferred over using the DSN6ARVP COMPACT parameter to compress archive logs written directly to tape since it:

• Removes the IDRC constraint  
• Facilitates disaster recovery by enabling you to take advantage of improved performance benefits provided by DFSMS compression

In addition, if you need faster log off-loading and the high performance that is available when reading striped archive logs, you can use Archive Log Accelerator and configure DFSMS to stripe your archive logs when they are created. Then, when you determine DB2 will not reference the archive log anymore (or that it will reference the logs infrequently) you can direct DFHSM to place the logs on tape.

Note: Compressing or striping the archive logs and then establishing a DFSMS management class to direct DFHSM to offload the logs to tape is the
preferred method of placing archive logs on tape because it allows you to
take advantage of DFSMS compression and data striping benefits enabled
by using Archive Log Accelerator.

Archive Log Accelerator also processes non-extended format archive logs that
currently reside on tape. In this situation, Archive Log Accelerator detects that the
archive log resides on tape and, if the Archive Log Accelerator Compress option is
selected, it will compress the archive log using its internal compression routine.

However, the drawbacks to using this method are that you must use non-extended
format archive logs and you cannot use DFSMS to compress or stripe the archive
logs—thus, recovery and archive log offload processing times might not be
comparable to those provided by DFSMS compression or striping.

Note: Archive Log Accelerator tape compression is only supported for
non-extended format archive logs that do not span more than one tape
cartridge. For more information about how Archive Log Accelerator
processes non-extended format archive logs, see "Non-extended format
archive log processing" on page 31.

Prerequisites

Make sure that you have the minimum hardware requirements described in
Hardware requirements and the minimum software requirements described in
Software requirements in place before you install Archive Log Accelerator. For
additional requirements and considerations, see the Program Directory for IBM DB2
Archive Log Accelerator for z/OS.

Hardware requirements

Archive Log Accelerator runs on a mainframe system and must meet the hardware
requirements needed to run IBM z/OS and DB2 6.1 and above.

To use DFSMS compression, you must have one of the following types of
controllers:
- Controllers that are ESCON-attached and support concurrent copy
- 3990-6 controllers
- 3990-3 controllers that are extended platform and ESCON-attached
- 3990-3 controllers that have the RAMAC support-level microcode
- 9394 controller
- 9343 controller with cache
- RAMAC

Software requirements

Archive Log Accelerator requires the following software:
- Any currently supported release of z/OS
- DB2 V6.1 or above
- SMS V1.2 or above (required for data striping and DFSMS compression). For
  more information, see DFSMS/MVS General Information and DFSMS/MVS
  Implementing System Managed Storage.
Chapter 2. Configuring Archive Log Accelerator

To install Archive Log Accelerator, follow the installation instructions provided in the Program Directory that was included with Archive Log Accelerator. After you have installed Archive Log Accelerator, perform the steps described in this chapter to configure Archive Log Accelerator for your environment.

In this chapter:
- “Configuration summary” on page 12
- “Step 1: Check authorization requirements” on page 12
- “Step 2: Create the DB2PARMS VSAM control file” on page 12
- “Step 3: Create the log file” on page 12
- “Step 4: Configure and initiate the started task” on page 13
- “Step 5: Configure the CLISTs and start the Archive Log Accelerator user interface” on page 14
- “Step 6: Configure DB2 subsystems” on page 16
- “Step 7: Configure Archive Log Parameters” on page 19
- “Step 8: Stop and restart the started task” on page 21
- “Step 9: (Optional) add parameters to started task JCL” on page 22
- “Step 10: (Optional) add Archive Log Accelerator to the DB2 Administration Tool Launchpad” on page 24
- “Configuring Archive Log Accelerator for a data sharing environment” on page 24

Configuration summary

Table 2 provides a summary of the steps you need to complete to configure Archive Log Accelerator for your system.

Table 2. Steps to configure Archive Log Accelerator

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Job</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check the authorities assigned to libraries and users.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Create the control file, if not previously created.</td>
<td>ALCCNTFL</td>
</tr>
<tr>
<td>3</td>
<td>Create the log file, if not previously created.</td>
<td>ALCLOGF</td>
</tr>
<tr>
<td>4</td>
<td>Configure and initiate the started task— and learn how to handle the started task during an IPL.</td>
<td>ALCPROC</td>
</tr>
<tr>
<td>5</td>
<td>Configure and run the CLIST to access to the main menu.</td>
<td>ALC, ALCCLIST</td>
</tr>
<tr>
<td>6</td>
<td>Specify each subsystem and libraries.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Specify archive log parameters for each subsystem.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Stop and restart the started task after specifying archive log parameters.</td>
<td>ALCPROC</td>
</tr>
<tr>
<td>Note: This step includes important information on what to do if the DB2 ZPARMS are modified.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>(Optional) Configure the started task JCL to override default processing.</td>
<td>ALCPROC</td>
</tr>
<tr>
<td>10</td>
<td>(Optional) Add Archive Log Accelerator to the DB2 Administration Tool launchpad.</td>
<td>ALCADBI</td>
</tr>
</tbody>
</table>
**Step 1: Check authorization requirements**

Archive Log Accelerator (ALC) requires the following authorities:

- The ALC load library must be APF authorized. If you have problems starting Archive Log Accelerator, contact your systems administrator.
- The ALC started task must have access (RACF ALTER authority) to read, write, delete, and rename DB2 archive log files.
- TSO users must have a minimum of READ authority to access the ZPARM member and the BSDS data sets.

**Step 2: Create the DB2PARMS VSAM control file**

The control file contains configuration information specific to Archive Log Accelerator. The JCL to create the control file resides in member ALCCNTFL in the Archive Log Accelerator sample library (SALCSAMP).

**Note:** Complete this step only if you do not already have a control file created by another DB2 product—or if you want to maintain a separate control file for Archive Log Accelerator.

To create the control file:
1. Add the appropriate job card to ALCCNTFL to meet your system requirements.
2. Change all occurrences of &CONTROL to the data set name for your installation's DB2 control file. For example: ALC.V220.CONTROL
3. Change &VOLSER to an appropriate volume serial number for your installation. For example: PCK001.
4. Submit ALCCNTFL to create the control file.

**Step 3: Create the log file**

A log file is required for Archive Log Accelerator. The log file records the actions taken by Archive Log Accelerator started task (the log file retains the last 256 compression event entries per DB2 subsystem). The following JCL for generating the log file is located in member ALCLOGF in the sample library (SALCSAMP).

```sql
LOGF EXEC PGM=IEFBR14
LOGFILE DD DSN=ALC.V220.LOGFILE,DISP=(NEW,CATLG,DELETE),RECFM=FB,
DSORG=PS,UNIT=SYSALLDA,
LRECL=80,BLKSIZE=16000,SPACE=(16000,(200,50))
```

**Table 3** shows the instructions for modifying the log file JCL:

<table>
<thead>
<tr>
<th>JCL Portion</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGFILE DD DSN</td>
<td>Change the high-level qualifier to meet your site's standards. In the example, the high-level qualifier is ALC.V220.</td>
</tr>
<tr>
<td>UNIT</td>
<td>Change the unit. In the example, the unit is SYSALLDA.</td>
</tr>
</tbody>
</table>

**Note:** If you created a log file for a previous version of Archive Log Accelerator, you can skip this step and use that same log file for this installation of Archive Log Accelerator.
Step 4: Configure and initiate the started task

The Archive Log Accelerator started task establishes and provides the operating environment required by Archive Log Accelerator. To configure and initiate the started task, complete the instructions in “Modify the started task JCL” and “Initiate the started task” on page 13.

Note: If the started task is not available, all requests to read a compressed archive log, or to decompress an archive log, can produce undesirable and potentially erroneous results.

Modify the started task JCL

Modify the JCL procedure for the started task from the SALCSAMP sample library, member ALCPROC, and then add it to MVS for started tasks initiation. The default procedure name is ALCPROC. The following is the started task JCL:

```
ALCPROC PROC
ALCPROC EXEC PGM=ALC#MAIN
STEPLIB DD DSN=ALC.V220.SALCLOAD,DISP=SHR
DB2PARMS DD DSN=ALC.V220.CONTROL,DISP=SHR
LOGFILE DD DSN=ALC.V220.LOGFILE,DISP=SHR
SYSPRINT DD SYSOUT=*  
```

You must configure the sample JCL to work in your environment. The JCL consists of parameters that are specific to your installation of DB2 and Archive Log Accelerator. The modifications you need to make to the started task JCL are described in Table 4.

Table 4. Started task JCL modifications

<table>
<thead>
<tr>
<th>JCL Portion</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB DD</td>
<td>Type the data set name of the installed load library for Archive Log Accelerator. In the sample JCL, the data set name is ALC.V220.SALCLOAD.</td>
</tr>
<tr>
<td>DB2PARMS DD</td>
<td>Type the data set name of the DB2PARMS VSAM control file. In the sample JCL, the data set name is ALC.V220.CONTROL. Note: The DB2PARMS data set might have been allocated when Archive Log Accelerator was installed, or by following the instructions in “Step 2: Create the DB2PARMS VSAM control file” on page 12</td>
</tr>
<tr>
<td>LOGFILE DD</td>
<td>Type the data set name of the log file. In the sample JCL, the data set name is ALC.V220.LOGFILE. Note: The LOGFILE data set might have been allocated when Archive Log Accelerator was installed, or by following the instructions in “Step 3: Create the log file” on page 12</td>
</tr>
<tr>
<td>EXEC</td>
<td>(Optional) If you are familiar with Archive Log Accelerator, you can add parameters to change the default processing. This optional step can be performed now or at a later time. For more information about using optional parameters in the started task, refer to “Step 9: (Optional) add parameters to started task JCL” on page 22</td>
</tr>
</tbody>
</table>

Initiate the started task

When you are finished modifying the JCL for the started task, you are ready to initiate the started task. To initiate the started task, type the MVS Start command, for example:

```
START ALCPROC  
```
This command example assumes that you retained the name of the started task as provided in the sample JCL.

IPLs and the automatic started task

It is recommended that you configure the Archive Log Accelerator started task to start automatically when MVS is IPLed. You can do this by adding the appropriate command to the COMMAND member within SYS1.PARMLIB (if needed, consult your MVS support personnel for assistance).

Notes:
1. The Archive Log Accelerator started task must be started on each MVS or z/OS image where DB2 instances exist for which Archive Log Accelerator is to process archive logs.
2. If you need to retain a previous version of Archive Log Accelerator, to avoid compatibility issues, make sure that you always start the most recent version of Archive Log Accelerator first, after an IPL.

Step 5: Configure the CLISTs and start the Archive Log Accelerator user interface

To access the Archive Log Accelerator user interface, you run the CLIST for Archive Log Accelerator. However, you must first configure the two CLISTS (ALC and ALCCLIST) found in the SALCSAMP sample library, to work in your environment.

ALC

The following is the sample JCL located in the ALC member of SALCSAMP:

```
PROC 0 CLISTLIB()
    CONTROL NOMSG
    FREE FILE(ALCCLIST)
    CONTROL MSG
    ALLOC FILE(ALCCLIST) DATASET('&CLISTLIB') SHR REU
    ALCCLIB ACTIVATE APPLICATION(CLIST) FILE(ALCCLIST)
    ISPEXEC SELECT CMD(ALCCLIST) NEWAPPL(ALC) PASSLIB
    ALCCLIB DEACTIVATE APPLICATION(CLIST)
    FREE FILE(ALCCLIST)
```

Instructions for modifying the sample JCL in the ALC member are provided in Table 5.

<table>
<thead>
<tr>
<th>Table 5. ALC member JCL modifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCL Portion</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>PROC 0 CLISTLIB()</td>
</tr>
</tbody>
</table>

ALCCLIST

The following is the second sample CLIST, ALCCLIST:

```
PROC 0 HILEVEL(ALC.V220)
    CONTROL NOMSG
    FREE FILE(DB2PARMS)
    FREE FILE(LOGFILE)
    CONTROL MSG
```
Instructions for modifying the sample JCL in the ALCCLIST member are provided in Table 6.

**Table 6. ALCCLIST member JCL modifications**

<table>
<thead>
<tr>
<th>JCL Portion</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROC 0 HILEVEL</td>
<td>Type the high-level qualifier between the parentheses. In the sample JCL, the line is PROC 0 HILEVEL(HILEVEL), where ALC.V220 is the high-level qualifier.</td>
</tr>
<tr>
<td>DB2PARMS</td>
<td>Type the data set name of the DB2PARMS VSAM control file between the parentheses. For example: ALLOC FILE(DB2PARMS) DA(HILEVEL.CONTROL). The sample JCL uses ALC.V220.CONTROL as the data set name.</td>
</tr>
<tr>
<td>LOGFILE</td>
<td>Type the data set name of the log file between the parentheses. For example: ALLOC FILE(LOGFILE) DA(HILEVEL.LOGFILE). In the sample JCL ALC.V220.LOGFILE is the data set name. <strong>Note:</strong> The LOGFILE referred to in the CLIST is the LOGFILE that the Archive Log Accelerator started task uses to record its activity. The LOGFILE designation should match the LOGFILE in use by the Archive Log Accelerator started task, or be the concatenation of all LOGFILE data sets used by any Archive Log Accelerator started task.</td>
</tr>
</tbody>
</table>

When you are finished modifying the ALCCLIST, copy the ALCCLIST member to your system CLIST library.
**Note:** A valid ISPF environment is required prior to running the ALCCLIST.

### Access the Archive Log Accelerator main menu

To access the Archive Log Accelerator main menu, type the following TSO command. For example:

`TSO ALC`

**Note:** This command example assumes that you retained the name of the CLIST as provided in the sample JCL.

The **DB2 Archive Log Accelerator for z/OS** main menu appears. Press Enter to remove the copyright statement.

```
----------- DB2 Archive Log Accelerator for z/OS -----------
Option ===>
  1 Current Status       User ID: myuserid
  2 Archive Log Reporting Date...: 2004/05/12
  S Setup and Configuration Time...: 21:15
                       Release: 0220
  A About Archive Log Accelerator
  X Exit Archive Log Accelerator

---
IBM* ROCKET**
Licensed Materials - Property of IBM 5697-I73
(C) Copyright IBM Corp. 2002,2004 All Rights Reserved.
(C) Copyright Rocket Software, Inc. 2001-2004 All Rights Reserved.
*Trademark of International Business Machines
**Trademark of Rocket Software, Inc
---
```

**Figure 4. DB2 Archive Log Accelerator for z/OS main menu**

### Step 6: Configure DB2 subsystems

The first time you run Archive Log Accelerator, you must set some general configuration options and then configure Archive Log Accelerator for each DB2 subsystem on which it will run. These configuration settings control the way in which Archive Log Accelerator runs and how it handles your DB2 archive log files. You control your archiving activity at the DB2 subsystem level, adapting your methods to meet your specific needs.

**Note:** If you are configuring Archive Log Accelerator for a data sharing environment, you must configure subsystems for each member of the data sharing group.

### Specify the DB2 subsystem

To specify the DB2 subsystem, follow these steps:

1. From the **DB2 Archive Log Accelerator for z/OS** main menu, type `S` (Setup and Configuration) and press Enter to access the **Enter DB2 System Parameters** screen.
The DB2 Control Dataset field displays the name of the DB2 VSAM control data set (this is the VSAM control data set described in “Step 2: Create the DB2PARMS VSAM control file” on page 12).

Note: This field is display only. You cannot change the contents of this field.

2. In the DB2 Subsystem ID field, type the 1 to 4 character subsystem ID of the DB2 subsystem that you want to configure. In Figure 5, the subsystem ID is ABCD.

Valid command selection values
The following options are available from the Enter DB2 System Parameters screen:

1. **ZPARM, BSDS, and Load Library Information**. Selecting this option displays the Update Parameters for DB2 Subsystem ssid screen. This screen enables you to input information specific to a particular DB2 subsystem, including ZPARM, BSDS, and load library information. For more information, see “Update parameters for the DB2 subsystem”.

2. **Archive Log Accelerator Parameters**. Selecting this option displays the Archive Log Configuration screen. This screen enables you to input information specific to Archive Log Accelerator (archive log parameters). For more information, see “Step 7: Configure Archive Log Parameters” on page 19.

3. **Delete Archive Log Accelerator Parameters for this DB2**. Selecting this option deletes archive log parameter settings for a specific subsystem. For more information, see “Deleting archive log parameters settings” on page 21.

After you specify the DB2 subsystem that you want to configure, input information specific to the subsystem as described in “Update parameters for the DB2 subsystem”.

Update parameters for the DB2 subsystem
To input information specific to a subsystem, type 1 (ZPARM, BSDS, and Load Library Information) in the Command line on the Enter DB2 System Parameters screen and press Enter. The Update Parameters for DB2 Subsystem ssid screen appears. For example:
This screen enables you to input information specific to a particular DB2 subsystem, including ZPARM, BSDS, and load library information. Enter the following subsystem parameters:

**DB2 ZPARMS Member**
Type the load module member name generated for this DB2 subsystem. In the example, the load module member is ABCDPARM.

**DB2 Bootstrap DSN #01 and #02**
Type the fully qualified data set names of the bootstrap data sets for this DB2 subsystem. In the example, the bootstrap data sets are ABCD.BSDS01 and ABCD.BSDS02.

**DB2 Loadlib1 through Loadlib5**
Type the names of the data sets that comprise the current load library concatenation for DB2. This concatenation is usually a subsystem-specific DSNEXIT library, the base DSNEXIT library for the current DB2 version, and the base DSNLOAD library for the current DB2 version. If necessary, two extra library spaces (DB2 Loadlib4 and DB2 Loadlib5) are provided.

*Note:* DB2 Loadlib1 through DB2 Loadlib5 must be the same libraries, specified in the same order, as the libraries used for the corresponding DB2 subsystem started task.

Press Enter to save the information, then press PF3 to exit this screen and return to the Enter DB2 System Parameters screen.

After you input information specific to a particular DB2 subsystem, configure the archive log parameters specific to Archive Log Accelerator as described in "Step 7: Configure Archive Log Accelerator"

*Note:* If you have been using a previous version of Archive Log Accelerator, you must delete your archive log parameter settings using the Delete Archive Log Accelerator Parameters for this DB2 option available on the Enter DB2 System Parameters prior to configuring Archive Log Accelerator V2.2. See "Deleting archive log parameters settings on page 21" for more information.
Step 7: Configure Archive Log Parameters

To configure archive log parameters for Archive Log Accelerator, type 2 (Archive Log Accelerator Parameters) in the Command line on the Enter DB2 System Parameters screen and press Enter. The Archive Log Configuration screen appears. For example:

```
----- Archive Log Accelerator Configuration -----  
Option ===>

DB2 Subsystem ID:
Copy 1 .... ABCD.ARCHLOG1 Dsnzparm ARCPF1
Compress... N (Y/N) Tape Only
Optimize... N (Y/N) Disk Only
Copy 2 .... ABCD.ARCHLOG2 Dsnzparm ARCPF2
Compress... N (Y/N) Tape Only
Optimize... N (Y/N) Disk Only

Work Dataset Definitions for Optimize=Y
Work Prefix  ABCD.ARCHLOG#____________________ DFSMS Compact=Y
Units...... BLK BLK,TRK,or CYL
Primary ... 001826 Primary Amount
Secondary.. 000365 Secondary Amount
Unit Name.. SYSALLDA Unit Esoteric

Press ENTER to save, END to exit. Press F1 for HELP.
```

Figure 7. Archive Log Accelerator Configuration screen

From the Archive Log Accelerator Configuration screen you can input information (archive log parameters) specific to Archive Log Accelerator. The archive log parameters enable you to adapt archive log management to your specific needs.

To help you determine your archive log parameter settings and understand the results you will obtain, see Determining archive log parameter settings on page 32. For example archive log scenarios, see Chapter 4, “Archive Log Scenarios,” on page 43.

Note: If you are configuring Archive Log Accelerator for a data sharing environment, you must configure archive log parameters for each member of the data sharing group.

Specify Copy 1 and Copy 2 archive log parameters

Copy 1 and Copy 2 are the archive logs that are produced by DB2. The Copy 1 and Copy 2 archive log parameters determine how these copies of the archive log are processed.

Notes:
1. If your installation does not produce a second copy of the archive log, any specifications for Copy 2 are ignored.
2. At a minimum, you must configure the Copy 1 parameters in order to use Archive Log Accelerator for the specified DB2 subsystem.

Specify the following Copy 1 and Copy 2 archive log parameters:

**Copy 1 and Copy 2**

The information in the Copy 1 and Copy 2 fields is obtained from DB2.
ZPARMS and is display only. If your installation does not produce a second copy of the archive log, the Copy 2 field is blank.

Note: If the data set name prefix that appears in this field has been defined as extended format by your storage administrator, the Compress parameter is ignored and Archive Log Accelerator allows SMS to process the archive log as defined by your storage administrator.

Compress

(For archive logs to be placed on tape.) Controls whether or not the archive log is compressed using Archive Log Accelerator’s compression routine when placing archive logs on tape. Type Y to compress the archive log using Archive Log Accelerator’s compression routine, otherwise, type N to leave the archive log uncompressed.

Note: For extended format archive logs, the Compress parameter is ignored and Archive Log Accelerator allows SMS to process the archive log as defined by your storage administrator.

Optimize

(For archive logs to be placed on disk.) Controls whether or not a DFSMS compressed archive log is optimized after compression. Type Y to optimize the archive log, otherwise, type N if you do not want to optimize the archive log.

Note: If the archive log will be optimized, you must specify Work Dataset Definitions as described in Specify work data set definitions (optimization only).

Specify work data set definitions (optimization only)

If you choose to optimize Copy 1 or Copy 2 of the archive log, you must specify work data set definitions. Archive Log Accelerator uses these work data set definitions to create a working copy of the archive log during the optimization process. For more information about how Archive Log Accelerator performs optimization, see "Understanding how Archive Log Accelerator processes archive logs" on page 27.

Work Prefix

This is the work data set name prefix. The data set name defaults to the data set name prefix specified in the DB2 ZPARMS where the last character of the name is replaced with pound sign (#). For example, if ABCD.ARCLOG1 has been specified in the DB2 ZPARMS, the default data set name would appear in this field as ABCD.ARCLOG#. You can override the contents by typing another data set name prefix name in this field, if needed.

Note: This work data set prefix must be defined for DFSMS compression by your storage administrator—otherwise, the archive log will not be optimized.

Units

How the work data set should be allocated. Specify one of the following values: BLK (Blocks), TRK (tracks), or CYL (cylinders). You can override the contents by typing another unit specification in this field, if needed.

Primary

The primary amount from DB2 ZPARMS. You can override the contents by typing another primary amount in this field, if needed.
Note: It is recommended that you use the default Archive Log Accelerator setting because the work data set is allocated in a manner to release all unused space when the optimize process is finished (thus, there is no wasted disk space).

Secondary
The secondary amount from DB2 ZPARMS. You can override the contents by typing another secondary amount in this field, if needed.

Note: It is recommended that you use the default Archive Log Accelerator setting because the work data set is allocated in a manner to release all unused space when the optimize process is finished (thus, there is no wasted disk space).

Unit Name
The esoteric unit name. You can override the contents by typing another unit name in this field, if needed.

When you are finished specifying the archive log parameters for this subsystem, press Enter to save the information and then press PF3 to exit and return to the Enter DB2 System Parameters screen.

Restoring archive log parameter default settings
To restore the default settings for the Archive Log Accelerator archive log parameters, type the DEFAULT, or DEF, command on the Option line. This command returns the settings for the Archive Log Configuration screen to their default values.

Deleting archive log parameters settings
To delete the archive log parameter settings for this DB2, select option 3 (Delete Archive Log Accelerator Parameters for this DB2) from the Enter DB2 System Parameters screen. Selecting this option deletes the archive log parameter settings by deleting the Archive Log Accelerator control record for the current DB2 subsystem.

Note: If you have been using a previous version of Archive Log Accelerator, you must delete your archive log parameter settings using the Delete Archive Log Accelerator Parameters for this DB2 option. After selecting this option, you must then select option 2 (Archive Log Accelerator Parameters) to reconfigure Archive Log Accelerator parameters for the DB2.

Step 8: Stop and restart the started task
Important: After you have specified the archive log parameters on the Archive Log Configuration screen, you must stop and restart the started task in order for the settings to take effect.

Stopping the started task
To stop the started task, type the following MVS MODIFY command. For example: MODIFY ALCPROC,STOP

This command example assumes that you retained the name of the started task as provided in the sample JCL. If you changed the name of the started task, you must use that name with the MVS MODIFY command to stop the task.
Notes:
1. If you decide not to stop and restart the started task at this time, the new configuration settings will take effect the next time the started task is stopped and started.
2. You should avoid using the MVS FORCE command to terminate a Archive Log Accelerator subsystem. Unpredictable results affecting DB2 and MVS could occur.

DB2 ZPARMS modifications and the started task

Important: If you modify the DB2 ZPARMS ARCPFX1 and ARCPFX2 values that define the data set prefix high-level qualifiers for the DB2 archive logs, you must perform the following steps for each DB2 subsystem that has had modifications to these DB2 ZPARMS values:

1. Issue the DEFAULT command to return your configuration settings to their default values. Returning your configuration settings to their default values is necessary because the Archive Log Accelerator DB2 subsystem parameter settings are saved in the DB2PARMS VSAM control file referenced by the Archive Log Accelerator started task JCL. To issue the DEFAULT command, type DEFAULT (or DEF) on the Option line of the Archive Log Configuration screen (accessible from option S, Setup and Configuration, on main menu, then option 2) and then press Enter.

Notes:
   a. It is recommended that you create a screen print of your configuration settings before issuing the DEFAULT command so you can refer to them when you want to configure these settings in the future.
   b. If you want to use archive log parameter settings other than the default settings provided by Archive Log Accelerator, you must also access the Archive Log Configuration screen to configure your desired settings.
   c. If you want to use archive log parameter settings other than the default settings provided by Archive Log Accelerator, you must also access the Archive Log Configuration screen to configure your desired settings.

2. Stop the Archive Log Accelerator started task.
3. Restart the Archive Log Accelerator started task. (Archive Log Accelerator will detect the new DB2 ZPARM settings after you have restarted the started task.)

Failure to follow these steps can produce unpredictable results.

Step 9: (Optional) add parameters to started task JCL

The Archive Log Accelerator started task has optional parameters (INCLUDE, EXCLUDE, and NOAUTO) that allow you to alter its default processing. These parameters control the started task—which, in turn, control Archive Log Accelerator processing.

The default processing for Archive Log Accelerator is to:
- Process all DB2 subsystems in the configuration that you specify on the Enter DB2 System Parameters screen. (To access this screen, choose option S, Setup and Configuration, on the main menu.)
- Automatically compress DB2 archive logs, based on your settings on the Archive Log Configuration screen for each DB2 subsystem. (To access this screen, choose option S, Setup and Configuration, on the main menu, then type 2 in the Command line to select the Archive Log Accelerator Parameters option.)

The following is the sample JCL for the started task located in member ALCPROC of the Archive Log Accelerator sample library:
You change the default processing by supplying optional parameters in the started task on the EXEC line as described in Table 7.

Table 7. Optional started task parameters

<table>
<thead>
<tr>
<th>EXEC Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCLUDE(SSID, SSID, ...)</td>
<td>This parameter explicitly lists the DB2 subsystems that are defined in Setup and Configuration (option S on the main menu) on which Archive Log Accelerator will operate. This parameter applies to the started task. If you specify this parameter, any DB2 subsystems that are not included will be unavailable for use with Archive Log Accelerator.</td>
</tr>
<tr>
<td>EXCLUDE(SSID, SSID, ...)</td>
<td>This parameter explicitly lists the DB2 subsystems that are defined in Setup and Configuration (option S on the main menu) on which Archive Log Accelerator will not operate. This parameter applies to the started task. If you specify this parameter, any DB2 subsystems that are not excluded will be available for use with Archive Log Accelerator.</td>
</tr>
<tr>
<td>NOAUTO</td>
<td>This parameter explicitly prohibits Archive Log Accelerator from using its internal compression routine to automatically compress archive log files when they are placed on tape. When you specify NOAUTO, you can use the services provided through option 1, Current Status, and option 2, Archive Log Reporting, on the main menu to read uncompressed or previously compressed logs. For example, with NOAUTO, you can perform compression activities offline, such as preparing off-site tapes for shipment, while still maintaining your ability to transparently process compressed archive logs and have full use of the Current Status and Archive Log Reporting options.</td>
</tr>
</tbody>
</table>

Notes:

1. If you specify NOAUTO and the Archive Log Accelerator started task is active, a compressed archive log will be automatically decompressed when DB2, or any other entity, attempts to read the compressed log. (The Archive Log Accelerator started task must be active in order to allow the archive log to be read.)

2. Regardless of whether or not the compressed archive logs reside on tape or disk, specifying NOAUTO only allows the logs to be read. It does not control whether or not SMS compresses the archive logs since this is determined by the SMS constructs assigned to the archive log data sets by your storage administrator.

Note: The INCLUDE and EXCLUDE parameters are mutually exclusive—that is, you can only specify one of these parameters on the EXEC line (not both).

Example 1: Using the INCLUDE and NOAUTO parameters

In the following example, the EXEC line contains two parameters: INCLUDE and NOAUTO.
include limits processing to only DB2 subsystems V61A and V61C. No other
subsystems will be processed even if others are listed using the Setup and
Configuration option (S on the main menu) of Archive Log Accelerator.

- NOAUTO prevents Archive Log Accelerator automatic compression from
  occurring. However, you can use the services provided through option 1,
  Current Status, and Option 2, Archive Log Reporting, on the main menu to read
  uncompressed or previously compressed logs.

```plaintext
//ALCPROC PROC
//ALCPROC EXEC PGM=ALC#MAIN,PARM='INCLUDE(V61A,V61C),NOAUTO'
//STEPLIB DD DSN=ALC.V220.SALCLOAD,DISP=SHR
//DB2PARMS DD DSN=ALC.V220.CONTROL,DISP=SHR
//LOGFILE DD DSN=ALC.V220.LOGFILE,DISP=SHR
//SYSPRINT DD SYSOUT=*  
```

**Example 2: Using the EXCLUDE parameter**

In the following example, the EXEC line contains the EXCLUDE parameter.
EXCLUDE instructs Archive Log Accelerator to process every DB2 subsystem that
has been specified using the Setup and Configuration option (S on the main
menu), except for subsystem V61B, which is listed in the EXCLUDE parameter.

```plaintext
//ALCPROC PROC
//ALCPROC EXEC PGM=ALC#MAIN,PARM='EXCLUDE(V61B)'
//STEPLIB DD DSN=ALC.V220.SALCLOAD,DISP=SHR
//DB2PARMS DD DSN=ALC.V220.CONTROL,DISP=SHR
//LOGFILE DD DSN=ALC.V220.LOGFILE,DISP=SHR
//SYSPRINT DD SYSOUT=*  
```

**Step 10: (Optional) add Archive Log Accelerator to the DB2 Administration Tool Launchpad**

You can optionally add Archive Log Accelerator to the DB2 Administration Tool
Launchpad. The Launchpad provides a central location to launch one or more DB2
tools.

**Note:** You must have created the Launchpad table in order to add Archive Log
Accelerator to the DB2 Administration Tool Launchpad.

To add Archive Log Accelerator to the DB2 Administration Tool Launchpad:

1. Locate the sample install CLIST in the following library: (XXX.XXXX(ALCADBI).
   Where XXX.XXXX is the library where Archive Log Accelerator is stored.
2. Modify the sample CLIST (ALCADBI) by following the instructions contained in
   the CLIST.
3. Run the ALCADBI CLIST to add Archive Log Accelerator to the launchpad.

**Note:** For detailed information on how to enable an IBM DB2 tool for the
launchpad, see the **IBM DB2 Administration Tool for z/OS User's Guide and
Reference**.

**Configuring Archive Log Accelerator for a data sharing environment**

If you are implementing Archive Log Accelerator in a data sharing environment, note
the following:

- You must configure Archive Log Accelerator for each member in the data sharing
  group where you want to process archive logs. (To access the Archive Log
  Configuration screen, select option S, Setup and Configuration.)
• If you are implementing Archive Log Accelerator for a DB2 that is a member of a data sharing group, the Archive Log Accelerator started task must be active on all of the systems where each member of the data sharing group resides where you want to process archive logs.

Within a data sharing environment, the Archive Log Accelerator control file can be shared. The log file should be the same log file that is used by all started tasks in the data sharing group; however, if it is not the same, the log file should be included in the LOGFILE concatenation used in the CLIST to ensure that the reports show all data sharing member activity.
Chapter 3. Using Archive Log Accelerator

Archive Log Accelerator enables you manage your archive logs using a variety of methods. This chapter provides information about how Archive Log Accelerator processes archive logs and also provides information that can help you determine your archive log parameter settings based on your archive log management goals.

In this chapter:
- "Understanding how Archive Log Accelerator processes archive logs"
- "Determining archive log parameter settings" on page 32
- "Understanding archive log parameter settings and results" on page 39

Understanding how Archive Log Accelerator processes archive logs

You can use Archive Log Accelerator to manage extended format archive logs (log defined as striped or DFSMS compressible) or non-extended format archive logs. Figure 8 shows how Archive Log Accelerator processes both types of archive logs:

![Diagram of Archive Log Accelerator processing](image)

Figure 8. Overview of Archive Log Accelerator processing

As shown in Figure 8, Archive Log Accelerator first detects whether or not the archive log is an extended format data set. If the archive log data set name prefixes...
have been defined as extended format data sets using DFSMS, Archive Log Accelerator processes the archive logs as described in "Extended format archive log processing" on page 28, otherwise the archive logs are processed as described in "Non-extended format archive log processing" on page 31.

Extended format archive log processing

As shown in Figure 9, if the archive log data set name prefixes specified by the DSNZPARM ARCPFX1 and ARCPFX2 settings have been defined as extended format data sets using DFSMS, Archive Log Accelerator allows the archive logs to be striped or compressed as defined in DFSMS by the storage administrator.

Figure 9. Overview of extended format archive log processing
If the archive log is defined as extended format, Archive Log Accelerator first determines whether or not the archive log has been defined for data striping only.

**Archive logs defined for data striping only**

If the archive log has been defined for data striping only, processing is complete. Archive Log Accelerator enables SMS to stripe the archive log as defined by the SMS constructs assigned to the archive log data set by the storage administrator.

**Archive logs defined for data striping and DFSMS compression, or DFSMS compression only**

If an archive log has not been defined for data striping only (that is, the log is either defined for striping and compression— or defined for DFSMS compression only), Archive Log Accelerator determines whether or not optimization has been requested by evaluating the **Optimize** parameter on the **Archive Log Accelerator Configuration** screen.

- If **Optimize** is **Y**, Archive Log Accelerator determines if the log is greater than 5 MB in size. If so, SMS processes the log and then Archive Log Accelerator optimizes the archive log.

- If **Optimize** is **N**, processing is complete. Archive Log Accelerator enables SMS to compress the archive log as defined by the SMS constructs assigned to the archive log data sets by the storage administrator.

**Figure 10. Extended format archive log processing flow**
**Note:** A data set must have a primary allocation of at least 5 MB in order to allow SMS compression to occur. If the data set’s primary allocation does not meet this requirement, SMS will not compress the data set. See *DFSMs: Using Data Sets*, for more information.

**Optimization processing**
When using DFSMS to compress archive logs, you can choose to optimize the compressed log by selecting Archive Log Accelerator’s **Optimize** option. The optimization process further reduces the size of the compressed archive log by removing unnecessary records from the log.

**Note:** Optimization is only available for those archive logs that are defined for DFSMS compression. If the log is not defined for DFSMS compression, the log cannot be optimized.

When **Optimize** = **Y**, Archive Log Accelerator optimizes the log by performing the following actions after SMS compresses the archive log:

1. Copies the original archive log data set to a work data set.
2. Removes unnecessary UNDO records from the working copy of the archive log.
3. Deletes the original archive log data set.
4. Renames the working copy of the archive log using the original archive log name.

**Figure 11** shows an overview of the optimization process.
When re-allocating the archive log (Step 1), Archive Log Accelerator creates a working copy of the archive log as specified by the Work Dataset Definitions fields. To name the work data set, Archive Log Accelerator uses the data set name prefix specified by the DSNZPARM ARCPFX1 and ARCPFX2 settings where the last character of the data set name prefix becomes a pound sign (#). For example:

<table>
<thead>
<tr>
<th>If the DSNZPARM settings are:</th>
<th>Archive Log Accelerator allocates a work copy of the archive log using the prefix:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCPFX1=ssid.ARCHLOG1</td>
<td>ssid.ARCHLOG#</td>
</tr>
<tr>
<td>ARCPFX2=ssid.ARCHLOG2</td>
<td>ssid.ARCHLOG#</td>
</tr>
</tbody>
</table>

Where ssid indicates the DB2 subsystem ID.

**Notes:**

1. Your storage administrator must define the original archive log and the work data set as extended format data sets using SMS.
2. Your storage administrator must use the appropriate SMS constructs to configure the original archive log and work data set as DFSMS compressible. The attributes of the archive log and the working copy of the archive log must match. If the archive log and work data set are not defined in this manner, Archive Log Accelerator will not optimize the archive log.
3. You can override the default work data set name by typing another data set name in the Work Prefix field, if needed.

For information about defining data sets for data striping and DFSMS compression, see Appendix A, “Modifying SMS Constructs and ACS Routines,” on page 59.

**Non-extended format archive log processing**

If the archive log is not extended format, Archive Log Accelerator determines the type of device on which the archive log resides (disk or tape).
If Archive Log Accelerator determines that the archive log is to be placed on disk, processing is complete (the archive log is left intact).

If Archive Log Accelerator determines the archive log is to be placed on tape, it determines whether or not compression has been requested by evaluating the Compress parameter setting on the Archive Log Accelerator Configuration screen:

– If Compress=Y (compress the archive log), Archive Log Accelerator uses its internal compression routine to compress the archive log.
– If the Compress=N (do not compress the archive log), processing is complete and an uncompressed copy of the archive log is created.

Notes:
1. Archive Log Accelerator’s internal compression routine can only be used to compress archive logs that reside on tape, it cannot be used to compress archive logs that reside on disk.
2. If the archive log is compressed using Archive Log Accelerator’s internal compression routine, the compressed archive log must be completely contained on a single tape.

Determining archive log parameter settings

Archive Log Accelerator provides flexible options that enable you to manage your archive logs in a variety of ways. The information provided in this section can help you determine the appropriate archive log parameter settings to use based on how you want Archive Log Accelerator to manage your archive logs.

Use the information provided in Table 9 to help you locate the appropriate configuration information and user scenarios based on your specific archive log management goals.

<table>
<thead>
<tr>
<th>If you want to:</th>
<th>Refer to:</th>
<th>For a user scenario, see:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduce the amount of time it takes to write the archive log</td>
<td>“To accelerate the archive log offload process” on page 33</td>
<td>“Scenario 1: Striping archive logs to accelerate archive log offload processing” on page 44</td>
</tr>
<tr>
<td>Experience improved compression ratios and take advantage of hardware-based compression</td>
<td>“To accelerate DB2 data recovery” on page 35</td>
<td>“Scenario 2: Compressing archive logs to accelerate DB2 data recovery” on page 44</td>
</tr>
<tr>
<td>Conserve storage space by optimizing DFSMS compressed archive logs</td>
<td>“To optimize compression” on page 37</td>
<td>“Scenario 3: Optimizing compression to conserve storage space” on page 45</td>
</tr>
<tr>
<td>Experience improved compression ratios, take advantage of DFSMS hardware-based compression, and reduce the amount of time it takes to write the archive logs</td>
<td>“To accelerate the archive log offload process” on page 33 or “To accelerate DB2 data recovery” on page 35</td>
<td>“Scenario 4: Accelerating archive log offloading and DB2 data recovery” on page 47</td>
</tr>
</tbody>
</table>

Note: Either configuration example is valid when compressing and striping archive logs.
Table 9. Determining archive log parameter settings based on archive log management goals (continued)

<table>
<thead>
<tr>
<th>If you want to:</th>
<th>Refer to:</th>
<th>For a user scenario, see:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experience improved compression ratios, take advantage of DFSMS hardware-based compression, reduce the amount of time it takes to write the archive logs, and conserve storage space</td>
<td>&quot;To optimize compression&quot; on page 37</td>
<td>&quot;Scenario 5: Accelerating archive log offloading, DB2 data recovery, and conserving storage&quot; on page 48</td>
</tr>
</tbody>
</table>

To accelerate the archive log offload process

Striping archive logs improves I/O performance and reduces the archive log window. Therefore, archive logs can be written faster and processing is accelerated. When you leverage data striping for your archive logs, you let DB2 create the archive logs and then allow SMS to stripe the logs as defined by your storage administrator. You can stripe one, or both, copies of the archive log.

Note: For maximum logging throughput and better DB2 performance, it is recommended that you stripe the active logs as well as the archive logs. That is, if the archive log is being striped, then the active log should also be striped (and vice-versa).

Copy 1 archive log parameter settings

The Copy 1 archive log parameter settings apply to the primary archive log produced by DB2. If you want to stripe Copy 1, specify the following archive log parameters on the Archive Log Configuration screen:

Copy 1

To use DFSMS to stripe the Copy 1 archive log, your storage administrator must use SMS to define this data set name prefix as an extended format data set that is striped. For specific instructions, see Appendix A, “Modifying SMS Constructs and ACS Routines,” on page 59.

Compress

The Compress archive log parameter does not apply to extended format data sets. Therefore, if Archive Log Accelerator detects that the data set name prefix that appears in the Copy 1 field has been defined as extended format, this parameter is ignored.

Optimize

The Optimize archive log parameter controls whether or not the archive log is optimized. This parameter only applies to archive logs that are defined for DFSMS compression, therefore the Optimize parameter is ignored.

Copy 2 archive log parameter settings

The Copy 2 archive log parameter settings on the Archive Log Configuration screen apply to the secondary archive log produced by DB2. If you want to stripe Copy 2, specify the following archive log parameters on the Archive Log Configuration screen:

Note: If your installation does not produce a second copy of the archive log, the Copy 2 archive log parameter settings are ignored.

Copy 2

To use DFSMS to stripe the Copy 2 archive log, your storage administrator
must use SMS to define this data set name prefix as an extended format data set that is striped. For specific instructions, see Appendix A, "Modifying SMS Constructs and ACS Routines," on page 59.

**Compress**
The **Compress** archive log parameter does not apply to extended format data sets. Therefore, if Archive Log Accelerator detects that the data set name prefix that appears in the **Copy 2** field has been defined as extended format, this parameter is ignored.

**Optimize**
The **Optimize** archive log parameter controls whether or not the archive log is optimized. This parameter only applies to archive logs that are defined for DFSMS compression, therefore the **Optimize** parameter is ignored.

**Example**
**Figure 13** shows example archive log parameter settings for striped archive logs.

```
----- Archive Log Accelerator Configuration -----

Option ==> _______________________________________________________________

DB2 Subsystem ID: ____________________________

Copy 1 .... ABCD.ARCHLOG1 Dsnzparm ARCPFX1
Compress... N (Y/N) Tape Only
Optimize... N (Y/N) Disk Only

Copy 2 .... ABCD.ARCHLOG2 Dsnzparm ARCPFX2
Compress... N (Y/N) Tape Only
Optimize... N (Y/N) Disk Only

Work Prefix Definitions for Optimize=Y
Work Prefix ABCD.ARCHLOG# ____________________________ DFSMS Compact=Y
Units_____ BLK BLK,TRK,or CYL
Primary... 001826 Primary Amount
Secondary.. 000365 Secondary Amount
Unit Name.. SYSALLDA Unit Esoteric

Press ENTER to save, END to exit. Press F1 for HELP.
```

**Figure 13. Example archive log parameter settings for striped logs (accelerating archive log offload processing)**

Note the following archive log parameter settings in **Figure 13**:

**Copy 1 and Copy 2**
The **Copy 1** and **Copy 2** data set name prefixes (ABCD.ARCHLOG1 and ABCD.ARCHLOG2) have been defined for data striping by the storage administrator.

**Note:** If the **Copy 1** and **Copy 2** data set name prefixes have not been defined for data striping using the appropriate SMS constructs, the archive logs will not be striped.

**Compress**
The **Compress** option is valid only for those logs that will be placed on tape. Since the archive logs have been defined as extended format data sets, and extended format data sets must reside on disk, the **Compress** settings are ignored.

**Optimize**
Because the archive logs are not defined for DFSMS compression, the **Optimize** parameter is ignored.
Work Dataset Definitions
The work data set definitions are ignored since the archive logs are defined for data striping.

To accelerate DB2 data recovery
Archive Log Accelerator speeds DB2 data recovery (specifically, recovery log processing) by providing the ability to use DFSMS to compress your archive logs. When you use DFSMS to compress your archive logs, you let DB2 create the primary and secondary archive logs and then allow DFSMS to compress the logs as defined by your storage administrator. You can compress one or both copies of the archive log.

Copy 1 archive log parameter settings
The Copy 1 archive log parameter settings apply to the primary archive log produced by DB2. If you want to use DFSMS to compress Copy 1, specify the following archive log parameters on the Archive Log Configuration screen:

Copy 1
To use DFSMS to compress the Copy 1 archive log, your storage administrator must use SMS to define this data set name prefix as an extended format data set that is DFSMS compressible. For specific instructions, see Appendix A, "Modifying SMS Constructs and ACS Routines," on page 59.

Compress
The Compress archive log parameter does not apply to extended format data sets. Therefore, if Archive Log Accelerator detects that the data set name prefix that appears in the Copy 1 field has been defined as extended format, this parameter is ignored.

Optimize
The Optimize archive log parameter controls whether or not the archive log is optimized. To simply use DFSMS to compress the archive log, specify N. If you want to optimize the compression, see "To optimize compression" on page 37.

Copy 2 archive log parameter settings
The Copy 2 archive log parameter settings on the Archive Log Configuration screen apply to the secondary archive log produced by DB2. If you want to use DFSMS to compress Copy 2, specify the following archive log parameters on the Archive Log Configuration screen:

Note: If your installation does not produce a second copy of the archive log, the Copy 2 archive log parameter settings are ignored.

Copy 2
To use DFSMS to compress the Copy 2 archive log, your storage administrator must use SMS to define this data set name prefix as an extended format data set that is DFSMS compressible. For specific instructions, see Appendix A, "Modifying SMS Constructs and ACS Routines," on page 59.

Compress
The Compress archive log parameter does not apply to extended format data sets. Therefore, if Archive Log Accelerator detects that the data set name prefix that appears in the Copy 2 field has been defined as extended format, this parameter is ignored.
Optimize

The **Optimize** archive log parameter controls whether or not the archive log is optimized. To simply use DFSMS to compress the archive log, specify \textit{N}. If you want to optimize the compression, see "To optimize compression" on page 37.

Example

**Figure 14** shows example archive log parameter settings for DFSMS compressed logs.

<table>
<thead>
<tr>
<th>Option</th>
<th>----- Archive Log Accelerator Configuration -----</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Subsystem ID:</td>
<td></td>
</tr>
<tr>
<td>Copy 1 .... ABCD.ARCHLOG1</td>
<td>Dsnzparm ARCPFX1</td>
</tr>
<tr>
<td>Compress... N</td>
<td>(Y/N) Tape Only</td>
</tr>
<tr>
<td>Optimize... N</td>
<td>(Y/N) Disk Only</td>
</tr>
<tr>
<td>Copy 2 .... ABCD.ARCHLOG2</td>
<td>Dsnzparm ARCPFX2</td>
</tr>
<tr>
<td>Compress... N</td>
<td>(Y/N) Tape Only</td>
</tr>
<tr>
<td>Optimize... N</td>
<td>(Y/N) Disk Only</td>
</tr>
<tr>
<td>Work Dataset Definitions for Optimize=Y</td>
<td></td>
</tr>
<tr>
<td>Work Prefix</td>
<td>ABCD.ARCHLOG#</td>
</tr>
<tr>
<td>Units......</td>
<td>BLK</td>
</tr>
<tr>
<td>Primary ...</td>
<td>001826</td>
</tr>
<tr>
<td>Secondary..</td>
<td>000365</td>
</tr>
<tr>
<td>Unit Name..</td>
<td>SYSALLDA</td>
</tr>
<tr>
<td>DFSMS Compact=Y</td>
<td></td>
</tr>
<tr>
<td>BLK,TRK,or CYL</td>
<td></td>
</tr>
<tr>
<td>Primary Amount</td>
<td></td>
</tr>
<tr>
<td>Secondary Amount</td>
<td></td>
</tr>
<tr>
<td>Unit Esoteric</td>
<td></td>
</tr>
</tbody>
</table>

Press ENTER to save, END to exit. Press F1 for HELP.

**Figure 14. Example archive log parameter settings for DFSMS compressed logs (accelerating DB2 data recovery)**

Note the following archive log parameter settings in **Figure 14**:

**Copy 1 and Copy 2**

The **Copy 1** and **Copy 2** data set name prefixes (ABCD.ARCHLOG1 and ABCD.ARCHLOG2) have been defined for DFSMS compression by the storage administrator.

**Note:** If the **Copy 1** and **Copy 2** data set name prefixes have not been defined as DFSMS compressible using the appropriate SMS constructs, the archive logs will not be compressed.

**Compress**

The **Compress** option is valid only for those logs that will be placed on tape. Since the archive logs have been defined as extended format data sets, and extended format data sets must reside on disk, the **Compress** settings are ignored.

**Optimize**

The **Optimize** options are set to \textit{N}, therefore, neither copy of the archive log will be optimized.

**Work Dataset Definitions**

Work data set definitions are ignored since **Optimize** is set to \textit{N} for **Copy 1** and **Copy 2** (optimization has not been requested).
To optimize compression

To maximize your compression savings, you can choose to optimize DFSMS compressed archive logs. Optimizing DFSMS compressed archive logs further reduces the size of logs and reduces the amount of storage space required to retain the archive logs.

Note: The Log Analysis Tool for z/OS and the DSN1LOGP utility do not support optimized archive logs.

Copy 1 archive log parameter settings
The Copy 1 archive log parameter settings apply to the primary archive log produced by DB2. If you want to optimize Copy 1, specify the following archive log parameters on the Archive Log Configuration screen:

Copy 1
To optimize the Copy 1 archive log, your storage administrator must use SMS to define this data set name prefix as an extended format data set that is DFSMS compressible. For specific instructions, see Appendix A, “Modifying SMS Constructs and ACS Routines,” on page 59.

Compress
The Compress archive log parameter does not apply to extended format data sets. Therefore, if Archive Log Accelerator detects that the data set name prefix that appears in the Copy 1 field has been defined as extended format, this parameter is ignored.

Optimize
The Optimize archive log parameter controls whether or not the archive log is optimized. To optimize the archive log, specify Y.

Copy 2 archive log parameter settings
The Copy 2 archive log parameter settings on the Archive Log Configuration screen apply to the secondary archive log produced by DB2. If you want to optimize Copy 2, specify the following archive log parameters on the Archive Log Configuration screen:

Note: If your installation does not produce a second copy of the archive log, the Copy 2 archive log parameter settings are ignored.

Copy 2
To use DFSMS to compress the Copy 2 archive log, your storage administrator must use SMS to define this data set name prefix as an extended format data set that is DFSMS compressible. For specific instructions, see Appendix A, “Modifying SMS Constructs and ACS Routines,” on page 59.

Compress
The Compress archive log parameter does not apply to extended format data sets. Therefore, if Archive Log Accelerator detects that the data set name prefix that appears in the Copy 2 field has been defined as extended format, this parameter is ignored.

Optimize
The Optimize archive log parameter controls whether or not the archive log is optimized. To optimize the archive log, specify Y.

Work dataset definitions
The work dataset definitions define the work data set Archive Log Accelerator uses to optimized the archive log.
Work Prefix
This is the work data set name prefix. The data set name defaults to the
data set name prefix specified in the DB2 ZPARMS where the last character
of the name is replaced with pound sign (#). For example, if
ABCD.ARCHLOG1 has been specified in the DB2 ZPARMS, the default
data set name would appear as ABCD.ARCHLOG# in this field. You can
override the contents by typing another data set name prefix name in this
field, if needed.

Note: This work data set prefix must be defined for DFSMS compression
by your storage administrator—otherwise, the archive log will not be
optimized.

Units How the work data set should be allocated. Specify one of the following
values: BLK (Blocks), TRK (tracks), or CYL (cylinders). You can override the
contents by typing another unit specification in this field, if needed.

Primary The primary amount from DB2 ZPARMS. You can override the contents by
typing another primary amount in this field, if needed.

Secondary The secondary amount from DB2 ZPARMS. You can override the contents
by typing another secondary amount in this field, if needed.

Unit Name The esoteric unit name. You can override the contents by typing another
unit name in this field, if needed.

Example
Figure 15 shows example archive log parameter settings for optimized archive logs.

--- Archive Log Accelerator Configuration ---

Option ===>

DB2 Subsystem ID:
Copy 1 .... ABCD.ARCHLOG1 Dsnzparm ARCPFX1
Compress... N (Y/N) Tape Only
Optimize... Y (Y/N) Disk Only

Copy 2 .... ABCD.ARCHLOG2 Dsnzparm ARCPFX2
Compress... N (Y/N) Tape Only
Optimize... Y (Y/N) Disk Only

Work Dataset Definitions for Optimize=Y
Work Prefix ABCD.ARCHLOG# DFSMS Compact=Y
Units...... BLK, TRK, or CYL
Primary ... 001826 Primary Amount
Secondary... 000365 Secondary Amount
Unit Name.. SYSCALLDA Unit Esoteric

Press ENTER to save, END to exit. Press F1 for HELP.

Figure 15. Example archive log parameter settings for optimized archive logs (optimizing
DFSMS compression)

Note the following archive log parameter settings in Figure 15.

Copy 1 and Copy 2
The Copy 1 and Copy 2 data set name prefixes (ABCD.ARCHLOG1 and
ABCD.ARCHLOG2) have been defined for DFSMS compression by the
storage administrator.
Note: If the Copy 1 and Copy 2 data set name prefixes have not been defined as DFSMS compressible using the appropriate SMS constructs, DFSMS compression will not occur and the archive logs will not be optimized.

Compress

The Compress option is valid only for those logs that will be placed on tape. Since the archive logs have been defined as extended format data sets, and extended format data sets must reside on disk, the Compress settings are ignored.

Optimize

The Optimize options are set to Y, therefore, both archive logs (Copy 1 and Copy 2) will be optimized.

Work Dataset Definitions

The contents of the Work Prefix field is populated by a default data set name prefix based on the DB2 ZPARMS ARCPFX1 and ARCHPFX2 settings (where the last character of the data set name prefix is replaced with a pound sign). Because Optimize is set to Y for Copy 1 and Copy 2, Archive Log Accelerator will use the work data set definitions to create a temporary work data set to use during optimization processing. For more information about how Archive Log Accelerator performs optimization, see “Optimization processing” on page 30.

Understanding archive log parameter settings and results

Table 10 shows the Archive Log Configuration screen Copy 1 and Copy 2 archive log parameter settings and the corresponding results.

Note: The Compress parameter does not apply to extended format archive logs and the Optimize parameter does not apply to non-extended format archive logs.

<table>
<thead>
<tr>
<th>If Copy 1 or Copy 2 is:</th>
<th>If Compress is:</th>
<th>If Optimize is:</th>
<th>The result is:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended format</td>
<td>Y</td>
<td>N</td>
<td>If the archive log is defined for DFSMS compression only, the archive log is compressed using DFSMS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If the archive log is defined for striping only, the archive log is striped as defined by SMS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If the archive log is defined for compression and striping, the archive log is compressed and striped as defined by SMS.</td>
</tr>
</tbody>
</table>
Table 10. Copy 1 and Copy 2 archive log parameter settings and corresponding results (continued)

<table>
<thead>
<tr>
<th>If Copy 1 or Copy 2 is:</th>
<th>If Compress is:</th>
<th>If Optimize is:</th>
<th>The result is:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended format</td>
<td>Y</td>
<td>Y</td>
<td>If the archive log is defined for DFSMS compression only, the archive log is compressed using DFSMS and optimized by Archive Log Accelerator.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If the archive log is defined for DFSMS compression and striping, the archive log is compressed and striped using DFSMS, and optimized by Archive Log Accelerator.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If the archive log is defined for striping only, the archive log is striped as defined by SMS. The archive log is not optimized since optimization only applies to DFMS compressed archive logs.</td>
</tr>
<tr>
<td>Non-extended format</td>
<td>Y</td>
<td>Y</td>
<td>If the archive log will be placed on disk, processing is complete (the log is left intact).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If the archive log will be placed on tape, Archive Log Accelerator uses its internal compression routine to compress the archive log.</td>
</tr>
</tbody>
</table>
### Table 10. Copy1 and Copy 2 archive log parameter settings and corresponding results (continued)

<table>
<thead>
<tr>
<th>If Copy 1 or Copy 2 is:</th>
<th>If Compress is:</th>
<th>If Optimize is:</th>
<th>The result is:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-extended format</td>
<td>Y</td>
<td>N</td>
<td>If the archive log will be placed on disk, processing is complete (the log is left intact).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If the archive log will be placed on tape, Archive Log Accelerator uses its internal compression routine to compress the archive log.</td>
</tr>
<tr>
<td>Non-extended format</td>
<td>N</td>
<td>Y</td>
<td>If the archive log will be placed on disk, processing is complete (the log is left intact).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If the archive log will be placed on tape, an uncompressed copy of the archive log is created.</td>
</tr>
<tr>
<td>Non-extended format</td>
<td>N</td>
<td>N</td>
<td>If the archive log will be placed on disk, processing is complete (the log is left intact).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If the archive log will be placed on tape, an uncompressed copy of the archive log is created.</td>
</tr>
</tbody>
</table>
Chapter 4. Archive Log Scenarios

A standard DB2 installation, with no other archiving tools, allows you to create one or two uncompressed archive logs. When you add Archive Log Accelerator, you can achieve a wide variety of archiving possibilities. Depending upon your installation and archiving requirements, there are a number of scenarios that you can follow. The most common scenarios are described in this chapter, though there are many possible variations.

In this chapter:

- Scenario 1: Striping archive logs to accelerate archive log offload processing
- Scenario 2: Compressing archive logs to accelerate DB2 data recovery
- Scenario 3: Optimizing compression to conserve storage space
- Scenario 4: Accelerating archive log offloading and DB2 data recovery
- Scenario 5: Accelerating archive log offloading, DB2 data recovery, and conserving storage

Usage considerations

For all of the scenarios described in this chapter:

- It is assumed that your installation produces Copy 1 and Copy 2 of the archive log.
- All fields mentioned in the scenarios are located on the Archive Log Configuration screen. To access this screen from the Archive Log Accelerator main menu, select option S, Setup and Configuration and then type 2 to select Archive Log Accelerator Parameters.
- To implement data striping or DFSMS compression, no changes are necessary to DB2. The configuration is accomplished using DFSMS. You do not need to stop and restart DB2 to implement data striping or DFSMS compression.
- You must configure Archive Log Accelerator for the DB2 so that it can recognize the DB2 DSNZPARM settings for the archive log. If Archive Log Accelerator is not configured for the DB2, it will not be able to perform the reading of the archive logs for DB2.
- DB2 accesses disk archive logs using BDAM. If Archive Log Accelerator is not active when DB2 attempts to read a striped or DFSMS compressed archive log, it will fail with an abend S213-94 and DB2 will terminate.
- If the archive logs will be placed on tape, define a DFSMS management class to direct when DFHSM should offload the logs to tape and when the logs can be scratched and then deleted. Note the following:
  - To stripe and compress the archive logs and then offload the logs to tape using HSM, make sure that the DSNZPARM CATALOG parameter is set to YES for the archive logs you want to stripe, compress, and offload to tape.
  - You can also define the management class to include the management of BSDS copies that are created during the archive log offload process.
  - Archive Log Accelerator is required for any access to data while the archive log is on disk, or if DB2 needs the log after it has been migrated to tape. When the log is migrated to tape, DFHSM will place the log back on disk...
before allowing DB2 to access the data. (When the log is placed back on disk it is striped or compressed, and Archive Log Accelerator must be running in order to service the data access.)

See *DFHSM, General Information*, for more information about DFHSM.

---

**Scenario 1: Striping archive logs to accelerate archive log offload processing**

You want to implement data striping for archive logs to improve log archiving processing times.

**Implementation steps**

To implement this scenario:

1. Configure Archive Log Accelerator to add the DB2 subsystem for which you want to use DFSMS to stripe the archive logs. See Chapter 2, “Configuring Archive Log Accelerator,” on page 11. For example configuration settings, see “To accelerate the archive log offload process” on page 33.

2. Configure DFSMS to request data striping for the archive logs (Copy 1, Copy 2, or both). To do this your storage administrator must:
   - a. Specify the data class attribute, Data Set Name Type, as EXTENDED.
   - b. Define a storage class defined for data striping. The storage class must have the Sustained Data Rate (SDR) attribute set to approximately 4MB/sec per desired stripe. (SMS determines the number of stripes based on the value of the Sustained Data Rate attribute in the storage class.) An existing data class that has been defined for data striping can be used.

   For specific instructions, see “Enabling data striping for archive logs” on page 61.

Upon completion of the DFSMS configuration and implementation, the next DB2 archive log that is created will be striped.

**Processing description**

When DB2 writes a striped archive log, the following actions occur:

- DFSMS allocates the archive log as striped.
- DB2 writes the archive log.
- Archive Log Accelerator tracks the writing of the archive log.

When DB2 reads the striped archive log, the following actions occur:

- The DB2 DSNMSTR address dynamically allocates the archive log.
- When DB2 attempts to open the striped log, Archive Log Accelerator performs the reads for DB2 (upon a READ request, Archive Log Accelerator reads the block on behalf of DB2).
- DB2 performs the necessary actions (forward and backward recovery).

---

**Scenario 2: Compressing archive logs to accelerate DB2 data recovery**

You want to implement hardware compression to reduce the space required to hold archive log data, as well as improve buffering and caching, and reducing I/O rates.

**Implementation steps**

To implement this scenario:
1. Configure Archive Log Accelerator to add the DB2 subsystem where you want to use DFSMS to compress the archive logs (Copy 1, Copy 2, or both). See Chapter 2, “Configuring Archive Log Accelerator,” on page 11 for instructions.

2. To simply accelerate archive log offload processing, ensure that the **Optimize** option on the **Archive Log Configuration** screen is set to N. For example configuration settings, see “To accelerate DB2 data recovery” on page 35. To optimize the compression, see “Scenario 3: Optimizing compression to conserve storage space” on page 45.

3. Configure DFSMS to request compression for the archive logs. To do this your storage administrator must:
   a. Configure a data class defined for compression (the data class Data Set Name Type attribute must be defined as EXTENDED and the Compaction attribute must be Y(YES). An existing data class that has been defined for DFSMS compression can be used.
   b. Modify your installation’s DFSMS DATACLAS ACS routine to assign the archive log data set name prefixes to the data class created in step a.

   For specific instructions, see “Enabling DFSMS compression for archive logs” on page 60.

Upon completion of the DFSMS configuration and implementation, the next DB2 archive log that is created will be DFSMS compressed.

**Processing description**

When DB2 writes a DFSMS compressed archive log, the following actions occur:

- DFSMS allocates the archive log as DFSMS compressed.
- DB2 writes the archive log.
- Archive Log Accelerator tracks the writing of the archive log.

When DB2 reads the DFSMS compressed archive log, the following actions occur:

- The DB2 DSNMSTR address dynamically allocates the archive log.
- When DB2 attempts to open the compressed archive log, Archive Log Accelerator performs the reads for DB2 (upon a READ request, Archive Log Accelerator reads the block on behalf of DB2).
- DB2 performs the necessary actions (forward and backward recovery).

**Scenario 3: Optimizing compression to conserve storage space**

You want to implement hardware compression to reduce space required to hold archive log data, as well as improve buffering and caching, and reducing I/O rates.

**Implementation steps**

To implement this scenario:

1. Configure Archive Log Accelerator to add the DB2 subsystem where you want to optimize archive logs (Copy 1, Copy 2, or both). See Chapter 2, “Configuring Archive Log Accelerator,” on page 11 for instructions.

2. On the **Archive Log Configuration** screen:
   a. Specify **Optimize** = Y to request optimization.
   b. Specify the required **Work Dataset Definitions** for optimization.

For example configuration settings, see “To optimize compression” on page 37
3. Configure DFSMS to request DFSMS compression for the work data set as described in step 4.

4. Configure DFSMS to request compression for the archive logs you want to compress. To do this your storage administrator must:
   a. Configure a data class defined for compression (the data class Data Set Name Type attribute must be defined as EXTENDED and the Compaction attribute must be Y(YES). An existing data class that has been defined for DFSMS compression can be used.
   b. Modify your installation’s DFSMS DATACLAS ACS routine to assign the archive log data set name prefixes to the data class created in step a. For specific instructions, see “Enabling DFSMS compression for archive logs” on page 60.

Upon completion of the DFSMS configuration and implementation, the next DB2 archive log that is created will be an extended format data set defined for DFSMS compression. Archive Log Accelerator will then optimize the archive log by removing unneeded records.

**Processing description**

When DB2 writes the DFSMS compressed archive log, the following actions occur:
- DFSMS allocates the archive log as extended format.
- DB2 writes the archive log.
- Archive Log Accelerator tracks the writing of the archive log.
- When the archive log is completely written, Archive Log Accelerator allocates the work data set, which is defined by the storage administrator as DFSMS compressed.
- Archive Log Accelerator performs optimization by copying the archive log to the work data set and removing the unneeded records.
- Archive Log Accelerator deletes the original archive log.
- Archive Log Accelerator renames the work data set using the original archive log name.

When DB2 reads the DFSMS compressed archive log, the following actions occur:
- The DB2 DSNMSTR address dynamically allocates the archive log.
- When DB2 attempts to open the compressed archive log, Archive Log Accelerator performs the reads for DB2 (upon a READ request, Archive Log Accelerator reads the block on behalf of DB2).
- DB2 performs the necessary actions (forward and backward recovery).

**Notes:**
1. The original archive log and the work data set required for optimization must be defined as extended format data sets using SMS.
2. Your storage administrator must use the appropriate SMS constructs to configure the work data set name to be compressed. If the archive log is not defined in this manner, optimization will not occur.
Scenario 4: Accelerating archive log offloading and DB2 data recovery

You want to accelerate archive log offloading in addition to accelerating DB2 data recovery. To do so, your storage administrator uses DFSMS to request compression for the archive logs in addition to data striping.

Note: This scenario is a combination of scenarios 1 and 2.

Implementation steps

To implement this scenario, complete the following steps:

1. Configure Archive Log Accelerator to add the DB2 subsystem where you want to use DFSMS to stripe and compress the archive logs (Copy 1, Copy 2, or both). See Chapter 2, “Configuring Archive Log Accelerator,” on page 11 for instructions.

2. To simply accelerate DB2 data recovery and archive log offload processing, ensure that the Optimize option on the Archive Log Configuration screen is set to N. For example configuration settings, see “To accelerate the archive log offload process” on page 33 or “To accelerate DB2 data recovery” on page 35.

Note: To optimize the compression, see “Scenario 5: Accelerating archive log offloading, DB2 data recovery, and conserving storage” on page 48.

3. Configure DFSMS to request striping and compression for the archive logs. To do this your storage administrator must:
   a. Configure a data class defined for compression (the data class Data Set Name Type attribute must be defined as EXTENDED and the Compaction attribute must be Y(YES). An existing data class that has been defined for DFSMS compression can be used.
   b. Modify your installation's DFSMS DATACLAS ACS routine to assign the archive log data set name prefixes to the data class created in step a.
   c. Define a storage class defined for data striping. The storage class must have the Sustained Data Rate (SDR) attribute set to approximately 4MB/sec per desired stripe. (SMS determines the number of stripes based on the value of the Sustained Data Rate attribute in the storage class.) An existing data class that has been defined for data striping can be used.

For specific instructions on modifying the SMS constructs required to enable data striping and DFSMS compression, see Appendix A, “Modifying SMS Constructs and ACS Routines,” on page 59.

Upon completion of the DFSMS configuration and implementation, the next DB2 archive log that is created will be DFSMS compressed and striped.

Processing description

When DB2 writes the archive log, the following actions occur:

- DFSMS allocates the archive log as extended format (striped and compressed).
- DB2 writes the archive log.
- Archive Log Accelerator tracks the writing of the archive log.

When DB2 reads the archive log, the following actions occur:

- The DB2 DSNMSTR address dynamically allocates the archive log.
- When DB2 attempts to open the archive log, Archive Log Accelerator performs the reads for DB2 (upon a READ request, Archive Log Accelerator reads the block on behalf of DB2).
Scenario 5: Accelerating archive log offloading, DB2 data recovery, and conserving storage

You want to conserve storage space while accelerating DB2 data recovery and archive log offloading. To do so, configure DFSMS to request compression for the archive logs in addition to data striping and configure Archive Log Accelerator to optimize the DFSMS compressed archive logs.

Note: This scenario is a combination of scenarios 1, 2, and 3.

Implementation steps

To implement this scenario, complete the following steps:

1. Configure Archive Log Accelerator to add the DB2 subsystem where you want to use DFSMS to stripe and compress the archive logs (Copy 1, Copy 2, or both). See Chapter 2, “Configuring Archive Log Accelerator,” on page 11 for instructions.

2. On the Archive Log Configuration screen:

   a. Specify Optimize=Y to request optimization.

   b. Specify the required Work Dataset Definitions for optimization.

   For example configuration settings, see "To optimize compression" on page 37.

3. Configure DFSMS to request DFSMS compression for the work data set as described in step 4.

4. Configure DFSMS to request striping and compression for the archive logs. To do this your storage administrator must:

   a. Configure a data class defined for compression (the data class Data Set Name Type attribute must be defined as EXTENDED and the Compaction attribute must be Y(YES). (An existing data class that has been defined for DFSMS compression can be used.)

   b. Modify your installation’s DFSMS DATACLAS ACS routine to assign the archive log data set name prefixes to the data class created in step a.

   c. Define a storage class defined for data striping. The storage class must have the Sustained Data Rate (SDR) attribute set to approximately 4MB/sec per desired stripe. (SMS determines the number of stripes based on the value of the Sustained Data Rate attribute in the storage class.) An existing data class that has been defined for data striping can be used.

   For specific instructions on modifying the SMS constructs required to enable data striping and DFSMS compression, see Appendix A, “Modifying SMS Constructs and ACS Routines,” on page 59.

Upon completion of the DFSMS configuration and implementation, the next DB2 archive log that is created will be DFSMS compressed and striped. In addition, Archive Log Accelerator will optimize the archive log.

Processing description

When DB2 writes the archive log, the following actions occur:

- DB2 performs the necessary actions (forward and backward recovery).

- DFSMS allocates the archive log as extended format (striped and compressed).

- DB2 writes the archive log.

- Archive Log Accelerator tracks the writing of the archive log.
• When the archive log is completely written, Archive Log Accelerator allocates the work data set, which is defined by the storage administrator as DFSMS compressed and striped.

• Archive Log Accelerator performs optimization by copying the archive log to the work data set and removing the unneeded records.

• Archive Log Accelerator deletes the original archive log.

• Archive Log Accelerator renames the work data set using the original archive log name.

When DB2 reads the archive log, the following actions occur:

• The DB2 DSNMSTR address dynamically allocates the archive log.

• When DB2 attempts to open the compressed archive log, Archive Log Accelerator performs the reads for DB2 (upon a READ request, Archive Log Accelerator reads the block on behalf of DB2).

• DB2 performs the necessary actions (forward and backward recovery).

Notes:

1. The original archive log and the work data set required for optimization must be defined as extended format data sets using SMS.

2. Your storage administrator must use the appropriate SMS constructs to configure the work data set name to be compressed. If the archive log is not defined in this manner, optimization will not occur.
Chapter 5. Viewing Archive Log Compression Status

After you set the configuration options for your DB2 subsystems, and stopped and restarted the started task, you can use the **Current Status** option to view the current status of archive log activity as well as the Archive Log Accelerator configuration settings for a specific DB2 subsystem.

**In this chapter:**
- [Accessing the Current Status screen](#)
- ["Viewing configuration settings for a specific DB2 subsystem" on page 53](#)

### Accessing the Current Status screen

The **Current Status** option helps you determine the level of activity that is related to your DB2 archive logs. For example:

- The number of archive log requests that have been processed by DB2
- The number of requests that DB2 has made to read the archive logs
- The date and time of the last DB2 archive log that was written

To view the current status of archive log activity, type `1` in the Option line of the **Archive Log Accelerator** main menu and press Enter. The **Current Status** screen appears. For example:

```
----------------------- Current Status ----------------------- 2004/05/21 17:47:30
Option ===>            
ALC Started Task: ALCPROC  Current Status: ACTIVE
  Task Started: 05/21/04 8:56:53
DB2 Archive Log Requests: 0  DB2 Archive Log Reads: 0

<table>
<thead>
<tr>
<th>Cmd</th>
<th>DB2</th>
<th>Status</th>
<th>Reason</th>
<th>ArchReq</th>
<th>ReadReq</th>
<th>OthRead</th>
<th>Last Archive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ABCD</td>
<td>ACTIVE</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EFGH</td>
<td>ACTIVE</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IJKL</td>
<td>INACTIVE</td>
<td>ZPARMS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MNOP</td>
<td>INACTIVE</td>
<td>ZPARMS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TALA</td>
<td>INACTIVE</td>
<td>ZPARMS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MSLA</td>
<td>INACTIVE</td>
<td>ZPARMS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

******************************************************************************* Bottom of Data *********************************************************************************
```

**Figure 16. Current Status screen**

**Note:** In a data sharing environment, the **Current Status** screen shows only the activity for the current data sharing member on the MVS image where the Archive Log Accelerator started task is running.

The following fields on the **Current Status** screen are display only.

**ALC Started Task**

The name of the started task used by Archive Log Accelerator for processing. If the Archive Log Accelerator started task is not active this field is blank.
Current Status
The status of the started task:
- **ACTIVE**—The started task is running.
- **INACTIVE**—The started task is not running.
- **NOAUTO**—The started task is active for decompression services only.

Task Started
The date (MM:DD:YY) and time (HH:MM:SS) the started task was initialized. If no Archive Log Accelerator started task is active this field is blank.

DB2 Archive Log Requests
The number of times that an archive log request was processed by DB2 since the started task was initialized. If noArchive Log Accelerator started task is active this field is blank.

DB2 Archive Log Reads
The number of times DB2 has requested a read of an archive log. If no Archive Log Accelerator started task is active this field is blank.

DB2 Status
The DB2 subsystem identifier.

Status
The status of the Archive Log Accelerator started task on the DB2 subsystem:
- **ACTIVE**—Archive Log Accelerator is enabled and allowed to process requests for the selected DB2 subsystem.
- **INACTIVE**—Archive Log Accelerator is prohibited from performing activities against the selected DB2 subsystem.
- **UNDEFINED**—No DB2 subsystem has been defined.

Reason
The reason for the status:
- **ADMIN**—No DB2 subsystem has been defined.
- **(blank)**—No reason is shown because the Archive Log Accelerator started task is active for the selected DB2 subsystem.
- **EXCLUDED**—Archive Log Accelerator is prohibited from performing compress and decompress activities against the selected DB2 subsystem because the DB2 subsystem was explicitly or implicitly excluded in the started task parameters.
- **ZPARMS**—The ZPARM module specified during Archive Log Accelerator setup and configuration is not accessible to the Archive Log Accelerator started task. This could indicate that the DB2 subsystem is defined on a different MVS image for which shared DASD is not available.

ArchReq
The number of times that an archive log request was processed by DB2 since the Archive Log Accelerator started task was initialized.

ReadReq
The number of times that DB2 has requested a read of an archive log.

OthRead
The number of times a non-DB2 request has been made to read an archived log.

Last Archive
The date and time that the last archive log was written.
Viewing configuration settings for a specific DB2 subsystem

To view the current configuration settings for a specific DB2 subsystem, type C in the Cmd field next to the appropriate DB2 subsystem on the Current Status screen. A display-only configuration screen appears where you can view the Archive Log Accelerator configuration settings for the selected DB2 subsystem.

Note: This screen is the same screen as the Archive Log Configuration screen except that the information is display only.

For descriptions of the fields on the configuration screen, see “Step 7: Configure Archive Log Parameters” on page 19.
Chapter 6. Obtaining Archive Log Reports

The Archive Log Reporting option on the Archive Log Accelerator main menu provides a history of the DB2 archive logs. Using the Archive Log Reporting option, you can select a subsystem and view a listing of its archive log data files.

Note: You can also use the DB2 DSN1LOGP utility to obtain information about compressed (un-optimized) archive logs. The Archive Log Accelerator started task must running in order to use DSN1LOGP with Archive Log Accelerator compressed (unoptimized) data sets. For information about using DSN1LOGP, see the IBM DB2 Utility Guide and Reference.

In this chapter:

• Viewing archive log summary information
• “Viewing compression history for a specific DB2 subsystem” on page 56
• “Viewing BSDS archive history” on page 57

Viewing archive log summary information

The Archive Log Summary screen shows a summary of archive log information listed by DB2 subsystem. To view archive log summary information, type 2 in the Option line of the Archive Log Accelerator main menu and press Enter. The Archive Log Summary screen appears. For example:

```
ALC$SUM 0220 --------- Archive Log Summary --------- 2004/05/21 16:57:12
ROW 1 OF 9

-----------------------------------------------------------------------------------------------
<table>
<thead>
<tr>
<th>Cmd</th>
<th>DB2</th>
<th>NumArch</th>
<th>NumComp</th>
<th>From:</th>
<th>To:</th>
<th>Low RBA</th>
</tr>
</thead>
<tbody>
<tr>
<td>JJJ</td>
<td>ABCD</td>
<td>846</td>
<td>330</td>
<td>12/08/03</td>
<td>18:40:50</td>
<td>05/11/04</td>
</tr>
<tr>
<td></td>
<td>NONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00000000</td>
</tr>
<tr>
<td>EFGH</td>
<td>471</td>
<td>250</td>
<td>12/10/03</td>
<td>12:01:34</td>
<td>05/11/04</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>3JKL</td>
<td>9</td>
<td>2</td>
<td>05/25/03</td>
<td>07:43:32</td>
<td>05/11/04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>08:22:39</td>
<td>00000000</td>
</tr>
<tr>
<td>MSLA</td>
<td>233</td>
<td>58</td>
<td>05/15/03</td>
<td>05:48:13</td>
<td>05/17/04</td>
<td>16:56:05</td>
</tr>
<tr>
<td></td>
<td>MSLB</td>
<td>451</td>
<td>10</td>
<td>05/31/03</td>
<td>03:00:50</td>
<td>05/14/04</td>
</tr>
<tr>
<td></td>
<td>3TALA</td>
<td>245</td>
<td>111</td>
<td>08/24/03</td>
<td>15:26:53</td>
<td>05/17/03</td>
</tr>
<tr>
<td></td>
<td>TALB</td>
<td>49</td>
<td>33</td>
<td>08/27/03</td>
<td>07:56:47</td>
<td>05/14/03</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15:46:06</td>
<td>00000000</td>
</tr>
</tbody>
</table>
-----------------------------------------------------------------------------------------------
```

Valid Line Commands: (Compression Detail, BSDS Detail)
Option ==> ________________________________

Figure 17. Archive Log Summary screen

The following fields on the Archive Log Summary screen are display only.

DB2  The ID of the DB2 subsystem.

NumArch  The number of archived logs.

NumComp  The number of compressed archived logs.

From  The first date (MM:DD:YY) and time (HH:MM:SS) found in the archived log history.
To the last date (MM:DD:YY) and time (HH:MM:SS) found in the archived log history.

**Low RBA**
The lowest RBA (relative byte address) found in the archived log history.

**High RBA**
The highest RBA (relative byte address) found in the archived log history.

**Low LRSN**
The lowest LRSN (log record sequence number) found in the archived log history.

**High LRSN**
The highest LRSN (log record sequence number) found in the archived log history.

**Viewing compression history for a specific DB2 subsystem**

To view compression history for a specific DB2 subsystem, type C in the **Cmd** field next to the DB2 subsystem on the **Archive Log Summary** screen that you want to view. The **Compression History** screen appears. For example:

```
ALC$HIST 0220 ----------- Compression History ----------- 2004/03/26 17:28:31
Option ==> DB2 Subsystem ID: ABCD
ROW 1 OF 142

                      Timestamp     BlksIn     BlksOut  ReqBy   Dataset Name
12/11/03  8:13:59     1440        226   ALCPROC ABCD.COMPLOG1.D01345.T081
12/11/03  8:14:01     1440        213   PASSTHRU ABCD.COMPLOG1.D01345.T081
12/11/03  8:13:59     1440        313   ABCDMSTR ABCD.ARCHLOG1.D01345.T081
11/26/03 15:46:11     1440        252   ABCDMSTR ABCD.ARCHLOG1.D01330.T154
11/26/03 14:38:47     1440        227   ALCPROC ABCD.COMPLOG1.D01330.T143
11/26/03 14:38:47     1440        467   PASSTHRU ABCD.COMPLOG1.D01330.T143
11/26/03 14:38:45     1440        321   ABCDMSTR ABCD.ARCHLOG1.D01330.T143
10/30/03  28:26        1440        227   ALCPROC ABCD.COMPLOG1.D01303.T002
10/30/03  28:29        1440        250   PASSTHRU ABCD.COMPLOG1.D01303.T002
10/30/03  28:23        1440        307   ABCDMSTR ABCD.ARCHLOG1.D01303.T002
10/09/03  1:03:34      332         183   ALCPROC ABCD.COMPLOG1.D01282.T020
10/09/03  1:03:36      1440        226   PASSTHRU ABCD.COMPLOG1.D01282.T020
10/09/03  1:03:23      332         183   ABCDMSTR ABCD.ARCHLOG1.D01282.T020
```

**Figure 18. Compression History screen**

The **Compression History** screen shows only those archive logs that have been compressed. In a data sharing environment, compression history statistics are provided for all members of the data sharing group where you have configured Archive Log Accelerator to compress archive logs.

The following fields on the **Compression History** screen are display only.

**Timestamp**
The date and time that the log was created.

**BlksIn** The size of the log file (in blocks) before compression.

**BlksOut** The size of the log file (in blocks) after compression.
**ReqBy**
How the compression request was initiated.

**Dataset Name**
The complete data set name of the compressed archive log file.

**Note:** For DFSMS compressed data sets, Archive Log Accelerator reports the physical size of the data set to include the compression dictionary that is stored with the data.

---

### Viewing BSDS archive history

To view bootstrap data for a specific DB2 subsystem, type a **B** in the **Cmd** field next to the DB2 subsystem on the **Archive Log Summary** screen. The **BSDS Archive History** screen appears. For example:

```
ALC$DBSD 0220  ------------ BSDS Archive History  ------------  2004/09/08  10:56:1
DB2 Subsystem ID: C71B

ROW 1 OF 875

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Copy</th>
<th>From:</th>
<th>To:</th>
<th>Low RBA</th>
<th>High RBA</th>
</tr>
</thead>
<tbody>
<tr>
<td>_</td>
<td>2</td>
<td>08/27/04 12:43:18</td>
<td>08/27/04 12:43:45</td>
<td>0002BF0AE000</td>
<td>0002C1260FFF</td>
</tr>
<tr>
<td>_</td>
<td>1</td>
<td>08/27/04 12:43:19</td>
<td>08/27/04 12:43:46</td>
<td>0002BF0AE000</td>
<td>0002C1260FFF</td>
</tr>
<tr>
<td>_</td>
<td>2</td>
<td>08/27/04 12:42:58</td>
<td>08/27/04 12:43:19</td>
<td>0002BCEE0000</td>
<td>0002BFD04FFF</td>
</tr>
<tr>
<td>_</td>
<td>1</td>
<td>08/27/04 12:42:58</td>
<td>08/27/04 12:43:19</td>
<td>0002BCEE0000</td>
<td>0002BFD04FFF</td>
</tr>
<tr>
<td>_</td>
<td>2</td>
<td>06/28/04 13:38:04</td>
<td>08/27/04 12:42:58</td>
<td>0002B9D0E000</td>
<td>0002B9E3DFFFF</td>
</tr>
<tr>
<td>_</td>
<td>1</td>
<td>06/28/04 13:38:03</td>
<td>08/27/04 12:42:58</td>
<td>0002B9D0E000</td>
<td>0002B9E3DFFFF</td>
</tr>
<tr>
<td>_</td>
<td>2</td>
<td>06/28/04 13:07:32</td>
<td>06/28/04 13:38:03</td>
<td>0002B9D29000</td>
<td>0002BAD20FFF</td>
</tr>
<tr>
<td>_</td>
<td>1</td>
<td>06/28/04 13:07:32</td>
<td>06/28/04 13:38:04</td>
<td>0002B9D29000</td>
<td>0002BAD20FFF</td>
</tr>
<tr>
<td>_</td>
<td>2</td>
<td>06/28/04 13:02:29</td>
<td>06/28/04 13:07:32</td>
<td>0002BAD25000</td>
<td>0002BAD20FFF</td>
</tr>
<tr>
<td>_</td>
<td>1</td>
<td>06/28/04 13:02:29</td>
<td>06/28/04 13:07:32</td>
<td>0002BAD25000</td>
<td>0002BAD20FFF</td>
</tr>
<tr>
<td>_</td>
<td>2</td>
<td>06/28/04 12:45:16</td>
<td>06/28/04 13:02:29</td>
<td>0002BAD20000</td>
<td>0002BAD24FFF</td>
</tr>
<tr>
<td>_</td>
<td>1</td>
<td>06/28/04 12:45:16</td>
<td>06/28/04 13:02:28</td>
<td>0002BAD20000</td>
<td>0002BAD24FFF</td>
</tr>
<tr>
<td>_</td>
<td>2</td>
<td>06/11/04 17:44:56</td>
<td>06/28/04 12:45:16</td>
<td>0002B935B000</td>
<td>0002BAD1FFFF</td>
</tr>
<tr>
<td>_</td>
<td>1</td>
<td>06/11/04 17:44:56</td>
<td>06/28/04 12:45:16</td>
<td>0002B935B000</td>
<td>0002BAD1FFFF</td>
</tr>
<tr>
<td>_</td>
<td>2</td>
<td>06/11/04 17:42:22</td>
<td>06/11/04 17:44:56</td>
<td>0002B719B000</td>
<td>0002B935AFFF</td>
</tr>
</tbody>
</table>
```

*Figure 19. BSDS Archive History screen*

This **BSDS Archive History** screen displays the bootstrap data for the selected subsystem.

The following fields on the **BSDS Archive History** screen are display only.

- **Copy** The archive log copy (Copy 1 or Copy 2).
- **From** The first date and time found in the archived log.
- **To** The last date and time found in the archived log.
- **Low RBA** The lowest RBA (relative byte address) found in the archived log.
- **High RBA** The highest RBA (relative byte address) found in the archived log.
- **Low LRSN** The lowest LRSN (log record sequence number) found in the archived log.
- **High LRSN** The highest LRSN (log record sequence number) found in the archived log.
| Dataset Name | The complete data set name of the archive log file. |
| Volume       | The volume on which the archived log resides. For multi-volume archive logs, only the first volume is shown. |
| Unit         | The unit on which the archived log resides. |
Appendix A. Modifying SMS Constructs and ACS Routines

This appendix provides information to assist your storage administrator when defining the appropriate SMS constructs and automatic class selection (ACS) routines to enable DFSMS data striping and compression for DB2 archive logs.

Overview

SMS constructs (class and storage group definitions) specify attributes for DFSMS compression and data striping. ACS routines assign class and storage group definitions to data sets to enable them to be compressed or striped.

For example, Figure 20 shows how SMS constructs and ACS routines enable data compression for archive log data sets.

Note: Because DB2 dynamically allocates the archive logs, the ACS routines must assign the compression or striped attribute based on the data set prefixes specified using ARCPF1X1 and ARCPF2X2 of DSNZPARM.

Figure 20. Using SMS constructs and ACS routines to enable DFSMS compression for archive logs
For information about DFSMS and defining SMS constructs, see DFSMS/MVS V1R5 Implementing System-Managed Storage and z/OS V1R5.0 DFSMSdfp Storage Administration Reference.

**Enabling DFSMS compression for archive logs**

To implement DFSMS compression, your storage administrator must define the SMS constructs required for data compression and then modify your installation's ACS routines to assign these constructs as described in the following steps.

1. Using DFSMS, define the following SMS constructs:
   
   a. A data class that is defined for compression. DFSMS compression is controlled by a parameter in the data class definition. To configure a data class defined for compression, the data class must have the Data Set Name Type attribute defined as EXTENDED and the Compaction attribute set to Y(YES).

   **Figure 21** shows a sample of a data class (DB2COMP) defined for compression using ISMF:

   ![Figure 21](image)

   **Figure 21. Example data class (defined for compression)**

   b. A storage class to define storage requirements for the archive log data sets.

   c. A storage group to determine where the archive logs will be placed.

   **Note:** Existing classes and storage groups can be used.

2. Modify your installation's DFSMS DATACLAS ACS routine to assign the appropriate archive log data set prefixes (the ARCPFX1 and ARCPFX2 values configured in the DB2 subsystem DSNZPARM configuration) to the data class defined in step 1 so that the archive logs can be DFSMS compressed.

   **Figure 22** shows a sample DFSMS DATACLAS ACS routine that assigns all data sets that have a prefix of ABC1.ARCHLOG1 and ABC1.ARCHLOG2 to belong to a data class named DB2COMP. (DB2COMP is a data class that was defined using DFSMS to enable DFSMS compression.) ABC1.ARCHLOG1 and ABC1.ARCHLOG2 are the archive log prefixes specified in the DB2 DSNZPARMS for a subsystem named ABC1.
3. Modify your installation’s DFSMS STORCLAS ACS routine to include the storage class defined in step 1b.

4. Modify your installation’s DFSMS STORGRP ACS routine to include the storage group defined in step 1c.

5. Translate the modified ACS routines to translate the ACS routines into an executable form.

6. Validate the modified ACS routines to check for errors.

7. Activate the modified ACS routines to implement the modifications.

Once the ACS routines are activated, archive log data sets that have a data set name prefix assigned to the classes and groups, will be enabled for DFSMS compression.

**Note:** If a data set name of an archive log does not match the data set name assignment, those archive logs will not be enabled for DFSMS compression, and therefore, will not be DFSMS compressed.

---

### Enabling data striping for archive logs

To implement data striping, your storage administrator must define the SMS constructs required for data striping and then modify your installation’s ACS routines as described in the following steps.

1. Using DFSMS, define the following SMS constructs:
a. A data class. The data class determines if the data set should be allocated as striped. To configure a data class defined for data striping, the data class must have the Data Set Name Type attribute defined as EXTENDED.

b. A storage class defined for data striping. To define a storage class for data striping, the storage class must have the Sustained Data Rate (SDR) attribute set to approximately 4MB/sec per desired stripe. (SMS determines the number of volumes to use for a striped data set based on the value specified for the Sustained Data Rate attribute in the storage class. The value specified for the SDR is divided by 4 to obtain the number of stripes.) For example, if the SDR is set to 24, a storage group of 3390s would have a stripe count of 6 and the allocation would be spread across 6 volumes.

The following shows a sample storage class (DBSTRIPE) defined using ISMF:

```plaintext
CDS Name . . . . : ACTIVE
Storage Class Name : DBSTRIPE
DESCRIPTION : MAKE ARCHIVE LOGS HAVE STRIPE COUNT OF 4

Performance Objectives
Direct Millisecond Response . . . : 
Direct Bias . . . . . . . . . . : 
Sequential Millisecond Response : 
Sequential Bias . . . . . . . . : 
Initial Access Response Seconds : 
Sustained Data Rate (MB/sec) . . : 16
Availability . . . . . . . . . : NOPREF
Accessibility . . . . . . . : NOPREF
Backup . . . . . . . . . . . : 
Versioning . . . . . . . . : 
```

Figure 23. Sample storage class defined for data striping

c. A storage group to determine where the archive logs will be placed. It must be a pool type with at least as many volumes defined as the maximum number of stripes.

Note: Existing data and storage classes can be used.

2. Modify your installation’s DFSMS DATACLAS ACS routine to add the appropriate archive log data set prefixes (the ARCPFX1 and ARCPFX2 values configured in the DB2 subsystem DSNZPARM configuration) so that the archive logs can be striped.

Figure 24 shows a sample DFSMS DATACLAS routine that assigns all data sets that have the prefix ABC1.ARCHLOG1 and ABC1.ARCHLOG2 to belong to a DATACLAS called DBSTRIPE. (DBSTRIPE is a data class that was defined using DFSMS to enable data striping.) ABC1.ARCHLOG1 and ABC1.ARCHLOG2 are archive log prefixes specified in the DB2 DSNZPARMS for a subsystem named ABC1.
3. Modify your installation's DFSMS STORCLAS ACS routine to include the storage class defined in step 1 (the routine must be conditioned for striping based on the data set names as in the ACS DATACLAS routine).

4. Modify your installation's DFSMS STORGRP ACS routine to include the storage class defined in step 1 (to direct the data sets to the striping volumes).

5. Translate the modified ACS routines. Performing this action translates the ACS routine into an executable form.

6. Validate the modified ACS routines to check for errors.

7. Activate the modified ACS routines to implement the modifications.

Once the ACS routines are activated, the archive log data sets that have a data set name prefix assigned to the classes and groups will be enabled for data striping.

**Note:** If a data set name of an archive log does not match the data set name assignment, those archive logs will not be enabled for data striping, and therefore, will not be striped.

**Disabling DFSMS compression or data striping for archive logs**

If you are using DFSMS compression or data striping for your archive logs and you do not want to use compress or stripe specific archive logs, your storage administrator can “disable” DFSMS compression or data striping by modifying the ACS routines so that the appropriate SMS constructs are no longer assigned to these archive log data sets.
Appendix B. Using the Hardware Compression Modeling Tool

This appendix describes how to use Archive Log Accelerator’s hardware compression modeling tool to help you determine the compression options that best suit your needs.

About the hardware compression modeling tool (ALCMODEL)

Archive Log Accelerator provides a hardware compression modeling tool that enables you to measure the benefits of DFSMS compression with, and without, Archive Log Accelerator compression optimization.

The ALCMODEL member of the Archive Log Accelerator sample library contains the JCL for the hardware compression modeling tool. When you run the hardware compression modeling tool JCL, compress and optimize operations are performed on a copy of the input archive log you specify. (First, the copy of the archive log is compressed and statistics are collected, and then the copy is optimized and statistics are collected a second time.) Both sets of statistics are output to a modeling report that shows compression savings with and without optimization.

Note: The Archive Log Accelerator started task does not need be active to use the modeling tool. Using the modeling tool does not adversely impact system operations.

Modifying the ALCMODEL JCL

Describes how to modify and run the ALCMODEL JCL.

Modifying the ALCMODEL JCL

Figure 25 shows the contents of the ALCMODEL member.

```
//BCOMP EXEC PGM=ALCMODEL
//STEPLIB DD DSN=ALC.V220.LOADLIB,DISP=SHR
// AN EXISTING ARCHIVE LOG IN ARCHIN
//ARCHIN DD DISP=SHR,DSN=XXXX.ARCHLOG1.DYYDDD.THHMMSST.ANNNNNNN
// ARCHOUT: EXISTING DSN MUST BE ALLOCATED WITH COMPRESS=YES
// ARCHOUT DD DISP=SHR,DSN=XXXX.ARCHLOGO.DYYDDD.THHMMSST.ANNNNNNN
// SYSPRINT DD SYSOUT=*  
```

Figure 25. ALCMODEL member contents

Note the following items in the ALCMODEL JCL:

STEPLIB

Specifies the product load library.

Note: ‘ALC.V220’ is the suggested high-level qualifier.

ARCHIN DD

Specifies an existing DB2 archive log (the input archive log).

ARCHOUT DD

Specifies the data set name that will be assigned as DFSMS compressible (the output archive log).
SYSPRINT
   Specifies the destination of the modeling report.

Complete the following steps to modify and run the ALCMODEL member JCL:

1. DDNAME ARCHIN— Specify a valid name of an input archive log.
2. DDNAME ARCHOUT— Specify the name of the output archive log. Note the following:
   • If this is a new allocation, the DCB attributes and size must match those specified for DDNAME ARCHIN.
   • Your storage administrator must assign a DATAclas to the output archive log that will make the output file DFSMS compressed. Any valid method of assigning DFSMS compression is permitted. (If the data set already exists, it must be DFSMS compressed.)
3. Submit member ALCMODEL to run the JCL and obtain the report.

Sample report

Figure 26 shows an example of a report generated by the hardware compression modeling tool:
The report shows input (original) and output (model) archive log information, statistics for DFSMS compression, and statistics for optimized DFSMS compression.

### Original archive log information
This section of the report shows the original archive log information, specified by in the ALCMODE JCL (ARCHIN DD), that is used to compare against the model archive log.

**NAME**  The user specified name of the original archive log specified in the ALCMODE JCL.

**BLKSIZE**  The size of the original archive log (in blocks).

### Model archive log information
This section of the report shows information for the resulting model archive log (ARCHOUT DD) which shows potential compression and (or) optimization savings.

**NAME**  The name of the model output archive log.

---

**Figure 26. Sample hardware compression modeling tool report**

The report shows input (original) and output (model) archive log information, statistics for DFSMS compression, and statistics for optimized DFSMS compression.

<table>
<thead>
<tr>
<th><strong>NAME</strong></th>
<th>PDUSER.CYL500.CUSTOMER.LOG.A0000001</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BLKSIZE</strong></td>
<td>24576</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>NAME</strong></th>
<th>PDUSER.EFCOMP.CYL500.CUSTOMER.LOG.A0000001</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STRIPE</strong></td>
<td>092</td>
</tr>
</tbody>
</table>

---

**Performing compression modeling without optimization...**

**Model archive log summary for compression:**

**Without optimization:**
- Original size in bytes: 368664576
- Original size in blocks: 15001
- Compressed size in bytes: 124718126
- Compressed size in blocks: 5075
- Compression savings: 66.17%

**Performing compression modeling with optimization...**

**Model archive log summary for optimized compression:**

**With optimization:**
- Original size in bytes: 368664576
- Original size in blocks: 15001
- Compressed size in bytes: 122668188
- Compressed size in blocks: 4992
- Compression savings: 66.72%

Optimization compressed ARCHOUT an additional

**Bytes** ............... 2049938

**Blocks** ............... 0082
STRIPES
The number of stripes of the model output archive log.

Model archive log summary for compression
This section shows statistics for DFSMS compression without Archive Log Accelerator optimization.

**ORIGINAL SIZE IN BYTES**
The size (in bytes) of the original archive log (ARCHIN).

**ORIGINAL SIZE IN BLOCKS**
The size (in blocks) of the original archive log (ARCHIN).

**COMPRESSED SIZE IN BYTES**
The size (in bytes) of the model archive log (ARCHOUT).

**COMPRESSED SIZE IN BLOCKS**
The size (in blocks) of the model archive log (ARCHOUT).

**COMPRESSION SAVINGS**
The total amount of savings. The savings is the comparison of the input archive log (ARCHIN) to the new model DFSMS compressed archive log (ARCHOUT) expressed as a percentage.

Model archive log summary for optimized compression
This section shows statistics for compression with Archive Log Accelerator optimization.

**ORIGINAL SIZE IN BYTES**
The size (in bytes) of the original archive log (ARCHIN).

**ORIGINAL SIZE IN BLOCKS**
The size (in blocks) of the original archive log (ARCHIN).

**COMPRESSED SIZE IN BYTES**
The size (in bytes) of the model archive log (ARCHOUT).

**COMPRESSED SIZE IN BLOCKS**
The size (in blocks) of the model archive log (ARCHOUT).

**COMPRESSION SAVINGS**
The total amount of savings. The savings is the comparison of the input DB2 archive log (ARCHIN) to the new model DFSMS compressed and optimized archive log (ARCHOUT) expressed as a percentage.

Optimization compressed archout an additional
This section shows the amount of savings gained if Archive Log Accelerator optimization is used.

**BYTES**
The amount saved (in bytes) if optimization is used.

**BLOCKS**
The amount saved (in blocks) if optimization is used.
Appendix C. Migrating to Archive Log Accelerator V2.2

This appendix provides migration information for prior Archive Log Accelerator users (Archive Log Compression Tool or Archive Log Accelerator V2.1 users). If you currently use one of these tools to manage your archive logs, and you want to migrate to Archive Log Accelerator V2.2, you can implement Archive Log Accelerator V2.2 as described in this appendix.

Migrating from a previous version of Archive Log Accelerator

Archive Log Accelerator V2.2 can process logs that were compressed by the compression routine provided with the DB2 Archive Log Compression Tool or Archive Log Accelerator V2.1, and those logs that you have striped or compressed using DFSMS.

Considerations and requirements

**Archive logs compressed by Archive Log Accelerator**

If you have been using a prior version of Archive Log Accelerator and you have archive logs that were compressed using the tool’s internal compression routine, Archive Log Accelerator V2.2 will automatically detect when an archive log was compressed in this manner and expand the data as it is being referenced.

**Archive logs that you want to stripe or compress using DFSMS**

If you have been using the Archive Log Compression Tool or Archive Log Accelerator V2.1 and you do not have any archive logs that were compressed using either product’s internal compression routine, simply install and configure Archive Log Accelerator V2.1 as outlined in the Migration steps.

**DB2PARMS VSAM file**

If you are sharing the same DB2PARMS VSAM file from the previous release of Archive Log Accelerator, you must delete the current archive log parameter settings and then reconfigure Archive Log Accelerator V2.2 since there is a new set of records for V2.2. See “Deleting archive log parameters settings” on page 21 for more information.

**Migration steps**

To migrate to Archive Log Accelerator V2.2, perform the following steps:

1. Define the appropriate SMS constructs to compress and (or) stripe your archive logs. See Appendix A, “Modifying SMS Constructs and ACS Routines,” on page 59 for more information.

2. To place the archive logs on tape, define a DFSMS management class to direct when HSM should offload the logs to tape and when the logs can be scratched and then deleted. Note the following:
   - To stripe the archive logs and then offload the logs to tape using HSM, make sure that the DSNZPARM CATALOG parameter is set to YES for the archive logs you want to stripe and offload to tape.
   - You can also define the management class to include the management of BSDS copies that are created during the archive log offload process.

3. Install and configure Archive Log Accelerator V2.2 for each DB2 subsystem where you want Archive Log Accelerator to manage archive logs. See the Program Directory for Archive Log Accelerator for z/OS and Chapter 2, “Configuring Archive Log Accelerator,” on page 11 for instructions.
Note: If you are sharing the same DB2PARMS VSAM file from the previous release of Archive Log Accelerator, you must re-configure Archive Log Accelerator V2.2 since there is a new set of records for V2.2. See “Update parameters for the DB2 subsystem” on page 17 for more information.

Related documentation

In addition to the installation information provided in the Program Directory for Archive Log Accelerator for z/OS, Table 11 provides references to topics you might find helpful when migrating to Archive Log Accelerator V2.2.

Table 11. Migration information within this user guide

<table>
<thead>
<tr>
<th>For information about:</th>
<th>Refer to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation and configuration</td>
<td>Chapter 2, “Configuring Archive Log Accelerator,” on page 11</td>
</tr>
<tr>
<td>DB2PARMS VSAM file</td>
<td>“Deleting archive log parameters settings” on page 21</td>
</tr>
<tr>
<td>Determining how you want to manage your archive logs</td>
<td>Chapter 3, “Using Archive Log Accelerator,” on page 27</td>
</tr>
<tr>
<td>Defining SMS data classes</td>
<td>Appendix A, “Modifying SMS Constructs and ACS Routines,” on page 59</td>
</tr>
</tbody>
</table>
Appendix D. Archive Log Accelerator Messages

This appendix documents the messages issued by Archive Log Accelerator. All messages generated by Archive Log Accelerator have a severity code printed as the last character of the message ID. The severity codes are described in Table 12.

Table 12. Message severity codes

<table>
<thead>
<tr>
<th>Severity code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Information only. No user action required.</td>
</tr>
<tr>
<td>W</td>
<td>Warning message. Results might not be as expected.</td>
</tr>
<tr>
<td>E</td>
<td>Error message. Some might be user-correctable, read the User Response to determine the course of action.</td>
</tr>
</tbody>
</table>

ALC002E INVALID SELECTION
Explanation: You have typed an invalid selection in the Option line.
User Response: Type a valid option or X to Exit.

ALC030E INVALID OPTION
Explanation: You have typed an invalid option in the Option line.
User Response: The only valid option is C, Configuration.

ALC032E INVALID OPTION
Explanation: You have typed an invalid option in the Option line.
User Response: Valid options are C, Compress History and B, BSDS Archive History.

ALC081E Work Dataset Prefix required for OPT=Y
Explanation: An invalid value has been specified for the Work Prefix field on the Archive Log Accelerator Configuration screen. To optimize the archive log, you must specify a work data set prefix.
User Response: Please specify a work data set prefix. For more information, see [Step 7: Configure Archive Log Parameters] on page 19.

ALC082E Space must be BLK, TRK, or CYL
Explanation: An invalid value has been specified for the Space field on the Archive Log Accelerator Configuration screen.
User Response: Please specify a valid value (BLK, TRK, or CYL). For more information, see [Step 7: Configure Archive Log Parameters] on page 19.

ALC083I Specify unit name for work dataset allocation
Explanation: A unit name has not been specified for the Unit Name field on the Archive Log Accelerator Configuration screen.
User Response: Please specify a valid unit name. For more information, see [Step 7: Configure Archive Log Parameters] on page 19.

ALC084E The Archive Log Accelerator parameters record for DB2 subsystem ssid was deleted
Explanation: This informational message indicates that the Archive Log Accelerator parameters for the DB2 subsystem specified by ssid, have been deleted.
User Response: To reconfigure the DB2 subsystem, see [Step 7: Configure Archive Log Parameters] on page 19.

ALC085E There is no Archive Log Accelerator parameters record for DB2 subsystem ssid
Explanation: This informational message indicates that the Archive Log Accelerator parameters for the DB2 subsystem specified by ssid, no longer exist.
User Response: To reconfigure the DB2 subsystem, see [Step 7: Configure Archive Log Parameters] on page 19.

ALC089E Please enter a Y or N
Explanation: You have typed an invalid selection.
User Response: Valid entries are Y (Yes) or N (no).

ALC0101E INVALID PARAMETER SPECIFIED
Explanation: An invalid form or keyword was used in the PARM= statement of the started task JCL.
User Response: Ensure that the syntax is correct and attempt to start the service task again.

**Explanation:** Archive Log Accelerator did not start due to a problem that occurred during product initialization.

User Response: Verify that sufficient storage exists for the product to start, that the Archive Log Accelerator library is APF authorized, and that the product administration process has been completed. If this does not remedy the problem, contact IBM customer support.

**Explanation:** Both the INCLUDE and EXCLUDE parameters were coded in the started task JCL.

User Response: None required.

**Explanation:** This message indicates the successful, orderly termination of the Archive Log Accelerator service address space and its associated functions.

User Response: None required.

**Explanation:** This message details the version, release date, release time, and debug information that is of use to customer support.

User Response: Print a copy of this message and have it ready if you need to contact IBM customer support.

**Explanation:** This message indicates that the NOAUTO parameter was coded in the started task parameters. NOAUTO forces the Archive Log Accelerator started task to bypass any attempt to compress an archive log but makes its services available for reading previously compressed logs.

User Response: None required.

**Explanation:** This is the header that accompanies the following ALC0110 messages.

User Response: None required.

**Explanation:** This message details the status of Archive Log Accelerator in regard to both the administrative parameters defined and the effect of the INCLUDE and EXCLUDE parameters. Potential values for status are:

- **ACTIVE** - Archive Log Accelerator is enabled and allowed to process requests for the selected DB2 subsystem.
- **EXCLUDED** - Archive Log Accelerator is prohibited from performing compress and decompress activities against the selected DB2 subsystem because the DB2 subsystem was explicitly or implicitly excluded in the started task parameters.
• FAILED - Automatic archive compression failed during initialization for the DB2 subsystem. If a DB2 subsystem is defined in the administrative portion of the product and the ZPARM module entered on the administrative panels cannot be found, initialization fails and this status is produced.

Potential values for reason are:
• “blanks” - No reason exists.
• PARMS - The status is due to either the INCLUDE or EXCLUDE parameter.
• ZPARMS - The ZPARM module, that is provided on the administrative panels, could not be found, or the administrative data was not saved.

User Response: None required.

ALC0111E  INVALID OPERAND FOR COMMAND
Explanation: An invalid command operand was included with a MODIFY command directed to the Archive Log Accelerator service address space.
User Response: Correct the command and attempt to enter it again.

ALC0112E  INVALID COMMAND
Explanation: An invalid command keyword was used with a MODIFY command directed to the Archive Log Accelerator service address space.
User Response: Correct the command keyword and attempt to enter it again.

ALC0113I  ALC DIAGNOSTIC DISPLAY:
Explanation: This informational message indicates that the following messages will display diagnostic information that is helpful to customer support.
User Response: None required.

ALC0114I  SDA ADDRESS XXXXXXXX
Explanation: The contents of this informational message are helpful to customer support.
User Response: None required.

ALC0115E  INVALID COMMAND SYNTAX
Explanation: This message indicates that the structure of the command directed to the Archive Log Accelerator service address space is incorrect.
User Response: Correct the syntax and resubmit the command.

ALC0120E  DATASPACE ALLOCATION ERROR - ABENDING
Explanation: An error occurred while trying to allocate the cached compression data space. The archive process has ended and an 0C3 ABEND has occurred.
User Response: Contact IBM customer support as soon as possible.

ALC0121E  STORAGE ALLOCATION FAILED - ABENDING
Explanation: An error occurred while trying to allocate required storage for cached compression. The archive process has ended and an 0C3 ABEND has occurred.
User Response: Contact IBM customer support as soon as possible.

ALC0129I  COMPRESSED: dataset
Explanation: This informational message indicates that the specified dataset has been successfully created and contains compressed archive log data.
User Response: None required.

ALC0131I  SEQUENTIAL: dataset
Explanation: This informational message indicates that the specified archive log dataset has been created as an uncompressed extended format data set with a stripe count that is greater than or equal to one.
User Response: None required.

ALC0132I  DFSMS COMPRESSED: dataset
Explanation: This informational message indicates that the specified archive log dataset has been compressed using DFSMS hardware compression.
User Response: None required.

ALC0133I  EF SEQUENTIAL: dataset
Explanation: This informational message indicates that the specified archive log dataset has been created as an uncompressed extended format data set with a stripe count that is greater than or equal to one.
User Response: None required.

ALC0135E  INVALID INTERNAL COMMAND RECEIVED
Explanation: An Archive Log Accelerator internal error has occurred.
User Response: Contact IBM customer support.
**ALC0139E**  FAILURE DURING COMMAND TASK INITIALIZATION

**Explanation:** The Archive Log Accelerator command task could not be properly initialized.

**User Response:** Attempt to restart the service task. If the command task should fail again, contact IBM customer support.

---

**ALC0141E**  ABEND PROCESSING IN DISK COMPRESSION EVENT

**Explanation:** Archive Log Accelerator detected an abend while attempting to optimize the log.

**User Response:** Review the abend code in the started task output and take appropriate action. If the problem persists, please contact IBM customer support.

---

**ALC0143E**  SEVERE ERROR. LOST JOURNAL ENTRY DUE TO FAILURE

**Explanation:** An Archive Log Accelerator internal error has occurred which resulted in the loss of a journal entry in the Archive Log Accelerator activity log.

**User Response:** If you receive this message, contact IBM customer support as soon as possible. Please have access to the job log associated with the failing task.

**Note:** Archive Log Accelerator remains active and capable of compressing archive logs even after this error has occurred.

---

**ALC0150E**  ERROR WHILE PROCESSING RENAME

**Explanation:** An existing DB2 archive log could not be deleted. Processing of the archive log was unsuccessful.

**User Response:** Verify that the Archive Log Accelerator started task has the authority to delete DB2 archive log data sets.

---

**ALC0151E**  ERROR WHILE PROCESSING DELETE

**Explanation:** During optimization, Archive Log Accelerator builds a work data set to optimize the archive log. Archive Log Accelerator uses the work file to contain the new (optimized) log. Archive Log Accelerator then deletes the original archive log using IDCAMS and RENAMES the work file (using IDCAMS) to the original archive log name. This message indicates that an error occurred during the DELETE portion of the optimization process.

**User Response:** Review the started task output (any errors in IDCAMS processing are copied to the started task output). Consult the IDCAMS message documentation for more information.

---

**ALC0155I**  DB2 ARCHIVE LOG ACCELERATOR COMMAND TASK TERMINATING

**Explanation:** This informational message indicates that the command task has started an orderly shutdown.

**User Response:** None required.

---

**ALC0160W**  Optimization Ignored. Reason <reason>

**Explanation:** The optimization request was ignored for one of the following reasons:

- ARCHIVE LOG IS TOO SMALL—To optimize the log it must be greater than 5 MB. This size constraint is a DFSMS restriction (DFSMS will not assign the compression attribute to a DSN that is being allocated if it is 5 MB or less). LRECL IS NOT VALID—Archive logs have an LRECL of 4096. When the optimization phase attempted to open the archive log it was found to have a different LRECL. Determine if the archive log is really a DB2 archive log, or if something has happened to the data set attributes. ARCHIVE LOG ALLOCATION FAILED—When the optimization phase allocated the log it failed. Refer to the accompanying Z/OS related messages for more information. UNABLE TO OPEN THE ARCHIVE LOG—When the optimization phase attempted to open the archive log it failed. Refer to the accompanying Z/OS related messages for more information.

**User Response:** Refer the explanation description for the appropriate user response.

---

**ALC0161I**  ARCHIVE LOG DSN: <dsname>

**Explanation:** This informational only message provides the input data set name being used in optimization.

**User Response:** None required.

---

**ALC0166E**  OPTIMIZATION FAILED: WORK DATASET NOT DFSMS COMPRESSED FORMAT

**Explanation:** Archive Log Accelerator terminated the optimization request because the work data set was not defined as DFSMS COMPRESSED.

**User Response:** The work data set prefix specified on the Archive Log Accelerator configuration screen must have the same DFSMS compression attribute as the original archive log.

---

**ALC0167I**  OPTIMIZATION REQUESTED FOR <dsname>

**Explanation:** This informational message indicates that a DB2 archive log has been created and Archive Log Accelerator has begun the optimization phase.
**Explanation:** This informational message provides the name of the work data set used during optimization.

**User Response:** None required.

**Explanation:** The Archive Log Accelerator started task is not currently running.

**User Response:** Restart the Archive Log Accelerator started task.

**Explanation:** This informational message indicates that the specified dataset has been successfully created and contains compressed archive log data.

**User Response:** None required.

**Explanation:** This informational message indicates that the specified dataset has been successfully created and contains archive log data that is not in a compressed format.

**User Response:** None required.

**Explanation:** The Archive Log Accelerator started task was attempting to load the module indicated by the specified module name, and it was missing.

**User Response:** Check the STEPLIB allocation to ensure that the product load library is installed.

**Explanation:** Archive Log Accelerator could not create its main control block, the ALC System Data Area (ALCSDA), because the started task (or creating job) was not APF authorized.

**User Response:** Check the STEPLIB allocation to ensure that the product load library is installed and APF authorized.

**Explanation:** Archive Log Accelerator could not create its main control block, the ALC System Data Area (ALCSDA), because there was insufficient storage for the control block. (ALCSDA requires approximately 512 bytes of CSA storage.)

**User Response:** Check the CSA Subpool 241 memory usage and look for possible storage shortage.

**Explanation:** Archive Log Accelerator anchors its main control block, the ALC System Data Area (ALCSDA), using the MVS Name Token Services facility. An error return code was detected using the Name Token Services facility.
User Response: Refer to the MVS Programming: Authorized Assembler Services Reference, Volume 2 (ENFREQ-IXGWRITE) for the IEANTCR return code.

ALC313E  ALCSDA DELETE FAILURE: APF AUTHORIZATION
Explanation: Archive Log Accelerator could not delete its main control block, the ALC System Data Area (ALCSDA), because it has detected an APF authorization failure.
User Response: Check the STEPLIB allocation to ensure that the product load library is installed and APF authorized.

ALC314E  ALCSDA DELETE FAILURE: IEANTDL RC return code
Explanation: Archive Log Accelerator anchors its main control block, the ALC System Data Area (ALCSDA), using the MVS Name Token Services facility. An error return code was detected using the Name Token Services facility.
User Response: Refer to the MVS Programming: Authorized Assembler Services Reference, Volume 2 (ENFREQ-IXGWRITE) for the IEANTDL return code.

ALC320E  OPEN ERROR DETECTED FOR DSNAMEx dataset
Explanation: An open failed when accessing dataset. For a DFSMS compressed archive log, no compression blocks in versus blocks out information is calculated.
User Response: Examine other error messages associated with the OPEN to determine the cause of the failure.

ALC321E  SVC INSTALLATION ERROR: NOT APF AUTHORIZED
Explanation: Archive Log Accelerator could not install its OPEN, OPEN TYPE=J, or CLOSE SVCs because it detected an APF authorization failure.
User Response: Check the STEPLIB allocation to ensure the product load library is installed and APF authorized.

ALC322E  SVC INSTALLATION ERROR: NO STEPLIB
Explanation: Archive Log Accelerator SVC installation processing failed due to a missing STEPLIB DDNAME.
User Response: Ensure that the Archive Log Accelerator started task has a STEPLIB containing the APF authorized product load library.

ALC323E  SVC INSTALLATION ERROR: LOAD ERROR
Explanation: Archive Log Accelerator SVC installation processing failed due to a missing module.
User Response: Ensure that the Archive Log Accelerator started task has a STEPLIB containing the APF authorized product load library.

ALC324E  SVC INSTALLATION ERROR: SVC ROUTINE CSA AREA SIZE
Explanation: Archive Log Accelerator SVC installation processing failed due to a CSA shortage in Subpool 241.
User Response: Examine CSA 241 usage below the 16M line.
ALC325I SVC INSTALLATION: REPLACE DETECTED FOR module name

Explanation: This is an informational message showing that a product service, or SVC routine, was replaced either due to a maintenance upgrade or new release.

User Response: None required.

ALC326E SVC INSTALLATION ERROR: LOAD ALCSVCHK

Explanation: Archive Log Accelerator SVC installation processing failed due to a load failure for module ALCSVCHK.

User Response: Examine started task joblog for any CSV prefix messages to determine cause of load failure. A possible STEPLIB problem could have occurred.

ALC327E SVC INSTALLATION ERROR: ALCSVCHK SP241 STORAGE OBTAIN

Explanation: Archive Log Accelerator SVC installation processing failed due to a CSA storage OBTAIN failure in Subpool 241. Module ALCSVCHK requires 512 bytes of below the line CSA 241 storage in order to install.

User Response: Examine CSA SP241 below the 16M line storage usage for possible storage leaks.

ALC328E SVC INSTALLATION ERROR: SVCUPDTE RC return code

Explanation: Archive Log Accelerator SVC installation processing failed due to an error in the z/OS SVCUPDTE facility. The return code from SVCUPDTE is return code.

User Response: See the MVS Programming: Authorized Assembler Services Reference, Volume 4 (SETFRR-WTOR) for information on the SVCUPDTE return code.

ALC331E UNABLE TO OPEN FILE FOR QSAM DECOMPRESSION

Explanation: An OPEN error was detected for an archive log using QSAM read access.

User Response: Check the joblog for any OPEN error messages that have the prefix IEC*.

ALC340I INSTALL module name EP: address

Explanation: The module name was installed at location address.

User Response: None required.

ALC601E ALCMODEL FAILURE: MISSING DDNAME <ddname>

Explanation: The ALCMODEL report requires the following DDNAMES:

- SYSPRINT—The output ddname.
- ARCHIN —The input archive log.
- ARCHOUT —A work data set that matches the archive log attributes.

User Response: Ensure that the DDNAMES listed above have been properly specified in the ALCMODEL JCL. See Appendix B, "Using the Hardware Compression Modeling Tool," on page 65 for more information.

ALC602E ALCMODEL FAILURE: MODELING NOT SUPPORTED FOR TAPE ARCHIVE LOGS

Explanation: Tape archive logs are not supported with the modeling tool ALCMODEL.

User Response: Please specify an archive log that resides on disk (instead of tape).

ALC603E ALCMODEL FAILURE OPENING DDNAME <ddname>

Explanation: An error occurred opening the listed DDNAME.

User Response: See the accompanying Z/OS message for more information.

ALC901E DATASET NOT FOUND

Explanation: The Rocket Software default load library could not be located.

User Response: Ensure that the load library name is correct.

ALC902E NO DB2 SYSTEM NAME

Explanation: A DB2 subsystem ID must be specified for processing to occur.

User Response: Type a DB2 subsystem name in the DB2 Subsystem ID field.

ALC903E DATASET NOT FOUND

Explanation: The default GDG base data set name could not be located.

User Response: Ensure that the default GDG base data set name is correct.
<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Explanation</th>
<th>User Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALC904E</td>
<td>VSAM OPEN ERROR</td>
<td>The specified data set could not be opened for I/O.</td>
<td>There might be related WTO messages that were also issued as a result of this error. Attempt to determine the cause of the error from these other messages. If the problem persists, contact IBM customer support.</td>
</tr>
<tr>
<td>ALC905E</td>
<td>VSAM READ ERROR</td>
<td>An unexpected return code from VSAM was encountered while reading the control file.</td>
<td>There might be related WTO messages that were also issued as a result of this error. Attempt to determine the cause of the error from these other messages. If the problem persists, contact IBM customer support.</td>
</tr>
<tr>
<td>ALC906I</td>
<td>RECORD UPDATED</td>
<td>The control file record for DB2 subsystem ssid has been successfully updated.</td>
<td>None required.</td>
</tr>
<tr>
<td>ALC907E</td>
<td>ERROR IN VSAM UPDATE</td>
<td>An unexpected return code from VSAM was encountered while doing an update operation of the control file.</td>
<td>There might be related WTO messages that were also issued as a result of this error. Attempt to determine the cause of the error from these other messages. If the problem persists, contact IBM customer support.</td>
</tr>
<tr>
<td>ALC908I</td>
<td>RECORD ADDED</td>
<td>The control file record for DB2 subsystem ssid has been successfully added.</td>
<td>None required.</td>
</tr>
<tr>
<td>ALC909E</td>
<td>INVALID VALUE</td>
<td>An invalid value was typed where Archive Log Accelerator expects a Y or an N.</td>
<td>Type a Y or an N and press Enter.</td>
</tr>
<tr>
<td>ALC911E</td>
<td>INVALID UNIT</td>
<td>An invalid tape ID or DASD unit name was typed in the Unit field.</td>
<td>Type a valid tape ID or DASD unit name and press Enter.</td>
</tr>
<tr>
<td>ALC914E</td>
<td>INVALID DSN PREFIX</td>
<td>An invalid DSN prefix was provided that does not have the same number of qualifiers as the corresponding DB2 archive log DSN.</td>
<td>The DSN Prefix must have same number of qualifiers as the corresponding DB2 DSN prefix. Type a valid DSN prefix and press Enter.</td>
</tr>
<tr>
<td>ALC915E</td>
<td>INVALID DSN PREFIX</td>
<td>An invalid DSN prefix was typed that does not follow standard data set naming conventions.</td>
<td>Correct the DSN Prefix to follow standard data set naming conventions and press Enter.</td>
</tr>
<tr>
<td>ALC916E</td>
<td>INVALID VALUE</td>
<td>An invalid value was provided.</td>
<td>Type a valid value and press Enter.</td>
</tr>
<tr>
<td>ALC918E</td>
<td>THE DSN PREFIX MUST MATCH THE DB2 DSN PREFIX SPECIFIED IN DSNZPARM</td>
<td>A Data Set Name prefix was not provided for COPY 1. The DB2 must have at least a COPY 1 prefix configured in order to use Archive Log Accelerator for this DB2.</td>
<td>Specify a Data Set Name prefix for COPY 1.</td>
</tr>
<tr>
<td>ALC999E</td>
<td>COMPRESSION OVERFLOW - PLEASE CONTACT SUPPORT</td>
<td>An unexpected error has occurred during compression processing. Compression processing for the archive log has failed.</td>
<td>Contact IBM customer support.</td>
</tr>
</tbody>
</table>
Appendix E. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is as your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
J46A/G4  
555 Bailey Avenue  
San Jose, CA 95141-1003  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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

All IBM prices shown are IBM’s suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly
tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM’s application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name)(year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

### Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

<table>
<thead>
<tr>
<th>Term</th>
<th>Company</th>
<th>Other Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS/ESA</td>
<td>IBM</td>
<td>IBM</td>
</tr>
<tr>
<td>DB2</td>
<td>IMS</td>
<td>IMS</td>
</tr>
<tr>
<td>DB2 Universal Database</td>
<td>MVS</td>
<td>MVS/EESA</td>
</tr>
<tr>
<td>DFSMS/MVS</td>
<td>MVS/ESA</td>
<td>OS/390</td>
</tr>
<tr>
<td>DFSMSdfp</td>
<td>RACF</td>
<td>Redbooks</td>
</tr>
<tr>
<td>DFSMSHsm</td>
<td></td>
<td>System/390</td>
</tr>
<tr>
<td>DFSMSrmm</td>
<td></td>
<td>z/OS</td>
</tr>
</tbody>
</table>

Unix ® is a registered trademark of The Open Group in the United States and other countries.

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

This glossary includes definitions reprinted from the American National Dictionary for Information Processing Systems, various ISO publications, and the CCITT Eighth Plenary Assembly Red Book, Terms and Definitions. Definitions are identified as follows:

- Definitions reprinted from the American National Dictionary for Information Processing Systems, copyright 1982 by the Computer Business Equipment Manufacturers Association (CBEMA), are identified by the symbol (A) following the definition. Copies can be purchased from the American Standards Institute, 1430 Broadway, New York, New York 10018.

- Definitions reprinted from a published section of the International Organization for Standardization’s (ISO) Vocabulary—Information Processing or from a published section of the ISO Vocabulary—Office Machines are identified by the symbol (I) following the definition. Because many ISO definitions are also reproduced in the American National Dictionary for Information Processing Systems, ISO definitions may also be identified by the symbol (A).

- Definitions reprinted from working documents, draft proposals, or draft international standards of ISO Technical Committee 97, Subcommittee 1 (Vocabulary) are identified by the symbol (T) following the definition, indicating that final agreement has not yet been reached among its participating members.

- Definitions are also reprinted from the CCITT Eighth Plenary Assembly Red Book, Terms and Definitions and working documents published by the International Telegraph and Telephone Union, Geneva, 1985.

- Definitions that are specific to IBM products are so labeled.

The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposed or substantively different meaning.

Synonym with. This is a backward reference from a defined term to all other terms that have the same meaning.

See. This refers the reader to multiple-word terms that have the same last word.

See also. This refers the reader to terms that have a related, but not synonymous, meaning.

A

archive log. The portion of the DB2 log that contains log records that have been copied from the active log.

active log. The portion of the DB2 log to which log records are written as they are generated. The active log always contains the most recent log records, whereas the archive log holds those records that are older and no longer fit on the active log.

automatic class selection (ACS) routine. A procedural set of ACS language statements. Based on a set of input variables, the ACS language statements generate the name of a predefined SMS class, or a list of name of predefined storage groups, for a data set.

B

basic sequential access method (BSAM). An access method that OS/390 and z/OS uses for storing or retrieving data blocks in a continuous sequence, using either a sequential access or a direct access device.

bootstrap data set (BDS). A VSAM data set that contains name and status information for DB2 as well as RBA range specifications, for all active and archive log data sets. It also contains passwords for the DB2 directory and catalog, and lists of conditional restart and checkpoint records.

C

configuration. The process of describing to a system the devices, optional features, and program products that have been installed so that these features can be used.

data class. A collection of allocation and space attributes, defined by the storage administrator, that are used to create a data set.
**DFSMS/MVS.** An IBM System/390 licensed program that provides storage, data, and device management functions. When combined with MVS/ESA SP Version 5 it composes the base MVS/ESA operating environment. DFSMS/MVS consists of DFSMSdfp, DFMSdss, DFSMSHsm, and DFSMSrmm.

**E**

**extended format.** The format of a data set that has a data set name type (DSNTYPE) of EXTENDED. The data set is structured logically the same as a data set that is not in extended format but the physical format is different.

**L**

**log.** A collection of records that describe the events that occur during DB2 execution and that indicate their sequence. The information thus recorded is used for recovery in the event of a failure during DB2 execution.

**log record sequence number (LRSN).** A number that DB2 generates and associates with each log record. DB2 also uses the LRSN for page versioning. The LRSNs that a particular DB2 data sharing group generates from a strictly increasing sequence for each DB2 log and a strictly increasing sequence for each page across the DB2 group.

**Q**

**queued sequential access method (QSAM).** An extended version of the basic sequential access method (BSAM). When this method is used, a queue is formed of input data blocks that are awaiting processing or of output data blocks that have been processed and are awaiting transfer to auxiliary storage or to an output device.

**R**

**relative byte address (RBA).** The offset of a data record or control interval from the beginning of the storage space that is allocated to the data set or file to which it belongs.

**S**

**sequential data striping.** A type of data striping access specific to striped data sets.

**storage administrator.** A person in the data processing center who is responsible for defining, implementing, and maintaining storage management policies.

**storage class.** A collection of storage attributes that identify performance goals and availability requirements, defined by the storage administrator, used to select a device that can meet those goals and requirements.

**stripe.** In DFSMS/MVS, the portion of a striped data set that resides on one volume. The records in that portion are not always logically consecutive. the system distributes records among the stripes such that the volumes can be read from or written to simultaneously to gain better performance. Whether it is striped is not apparent to the application program.

**striped data set.** In DFSMS, an extended format data set consisting of two or more stripes. SMS determines the number of stripes based on the value of the SUSTAINED DATA RATE in the storage class.

**striping.** A software implementation of a disk array that distributes data across multiple volumes to improve performance.

**system management subsystem (SMS).** A DFSMS/MVS facility used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, management class, storage group, and ACS routine definitions.
Bibliography

This bibliography includes all the publications cited in this book,

- DFSMS/MVS Version 1 Release 2.0 Data Compression Implementation Guide, GG24-4251-00
- DFSMS/MVS General Information, GC26-4900-06
- DFSMS/MVS Version 1 Release 5 Implementing System Managed Storage, SC26-3123-05
- DFSMS/MVS Version 1 Release 5 Planning for Installation, SC26-4919-06
- OS/390 DFSMS Introduction, SC26-7344-00
- DFSMS/MVS Using Data Sets, SC26-4922-04
- DB2 Universal Database for OS/390 and z/OS Administration Guide Version 7, SC26-9931-01
- DFSMSdfp Storage Administration Reference, SC26-4920-05
- DFHSM 2.6.0 General Information, GH35-0092-07
Index

A
accelerating
archive log offload 33
DB2 data recovery 3, 35
offload processing 3
accelerating archive log offload scenario 44, 45, 47, 48
accelerating DB2 data recovery scenario 44, 47, 48
accessibility features xiii
active log 3
ALC sample JCL 14
ALC$DBSD screen 57
ALC$HIST screen 56
ALCCLIST sample JCL 14
ALCCNTFL 12
ALCLOGF sample JCL 12
ALCMODEL 65
ALCPNL2 screen 19
ALCSPROC sample JCL 13
Archive Log Accelerator
benefits 2
components 7
compression 32
compression optimizer 7
configuration settings 27
configuring 11, 16
control file 7, 12
environment 4
features 2
hardware compression modeling tool 7
ISPF interface 8
log file 12
parameters 19
performance considerations 9
prerequisites 10
scenarios 43
security considerations 8
started task 7, 13, 25
uses for 8
using 27
Archive Log Configuration screen 19
archive log parameters
Compress 20, 32
Copy 1 and Copy 2 19
deleting 21
Optimize 20
settings 32
settings and results 39
Work Prefix 20
archive logs 29
accelerating offload processing 3
and storage 1
compressing 35
data set prefixes 6
extended format 5, 28
history 55
archive logs (continued)
non-extended format 31
optimizing 3, 29, 37
processing 27
reporting 55
striping 29, 33
archiving scenarios 43
ARCPF1 28, 31
ARCPF2 28, 31
authorization requirements 12
B
benefits of using Archive Log Accelerator 2
BSDS
Archive History screen 57
information 17
C
CLIST 14
Compress parameter 20, 32
compressing 29
compressing archive logs 29, 35
scenario 44, 47, 48
compression
Archive Log Accelerator 32
DFSMS 3, 5, 28, 35, 37
Compression History screen 56
compression optimizer 7
configuration settings 32
for extended format archive logs 32
for non-extended format archive logs 32
configuring
Archive Log Accelerator 11
CLISTs 14
started task JCL 13
conserving storage
scenario 45, 48
control file
about 7
creating 12
Copy 1 and Copy 2
archive log parameters 19
D
data set
primary allocation 30
types supported 6
data sharing
and Current Status screen 51
and the started task 25
environment 6, 19, 25
data striping 29
implementing 5, 61
scenarios 44, 47, 48
DB2 Administration Tool 6
DB2 Administration Tool Launchpad 24
DB2 Change Accumulation Tool 6
DB2 data recovery
  accelerating 3, 35
DB2 Log Analysis Tool 6, 37
DB2 subsystems 16
DB2 ZPARMS 19
  and DEFAULT command 22
  and extended format data sets 6
  and the started task 22
  modifications 22
DB2PARMS data set 13
DB2PARMS VSAM control file 12
DEFAULT command 21
default processing 13, 22
default settings 21
deleting archive log parameters 21
DFSMS 28, 30
DFSMS compression
  implementing 5, 60
  optimizing 3
DSN1LOGP 6, 37
DSNZPARM 28, 31

E
Enter DB2 System Parameters screen 16
  environment
    Archive Log Accelerator 4
    data sharing 6, 19
EXCLUDE 23
  extended format
    archive logs 28, 32

H
hardware compression modeling tool 7, 65

I
implementing
  data striping 5, 61
  DFSMS compression 5, 60
INCLUDE 23
IPLs 14
ISPF interface 8

J
JCL
  ALC 14
  ALCCLIST 14
  ALCLOGF 12
  ALCSPROC 13
  started task 13

L
Launchpad, adding product to 24
  load library 17
  log file, creating 12
LOGFILE
  data set 13
  statement 25
LookAt message retrieval tool xii

M
main menu 16
message retrieval tool, LookAt xii
migration
  considerations 69
  steps 69
  modifications
    default processing 13
    to DB2 ZPARMS 22

N
NOAUTO 23
non-extended format
  archive logs 31
non-extended format archive logs 32
Notices 79

O
offload processing 3
Optimize parameter 20
  optimizing
    archive logs 3, 30, 37
    scenario 45, 48
  optimizing archive logs 29

P
performance considerations 9
prerequisites 10
processing
  archive logs 27
  optimization 30

Q
QSAM 3

R
requirements
  hardware 10
  software 10

S
sample JCL
  ALC 14
sample JCL (continued)
ALCCLIST  14
ALCSPROC  13

scenarios
accelerating archive log offload  44, 45, 47, 48
accelerating DB2 data recovery  44, 47, 48
archiving  43
compressing archive logs  44, 47, 48
conserving storage  45, 48
optimizing archive logs  45, 48
striping archive logs  44, 45, 47, 48

screen readers and magnifiers  xiii

Screens
ALC$DBSD  57
ALC$HIST  56
ALCPNL2  19
Archive Log Configuration  19
BSDS Archive History  57
Compression History  56
Enter DB2 System Parameters  16
main menu  16
Update Parameters for DB2 Subsystem  17

Secondary  21
security considerations  8

started task
and data sharing  25
and DB2 ZPARMS  22
and IPLs  14
Archive Log Accelerator  7, 13
initiating  13
JCL  13
parameters  22
restarting  21
stopping  21
striping
archive logs  29, 33

striping archive logs
scenario  44, 45, 47, 48

T
tape  32
TSO users  12

U
Unit Name  21
Units  20
Update Parameters for DB2 Subsystem screen  17

W
work data set definitions  37
  description  20
  Primary  20
  Secondary  21
  Unit Name  21
  Units  20
work dataset definitions  31
Work Prefix  20
Readers’ Comments — We’d Like to Hear from You

IBM DB2 Archive Log Accelerator for z/OS
User’s Guide
Version 2 Release 2

Publication No. SC18-7405-02

Overall, how satisfied are you with the information in this book?

<table>
<thead>
<tr>
<th>Overall satisfaction</th>
<th>Very Satisfied</th>
<th>Satisfied</th>
<th>Neutral</th>
<th>Dissatisfied</th>
<th>Very Dissatisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How satisfied are you that the information in this book is:

<table>
<thead>
<tr>
<th></th>
<th>Very Satisfied</th>
<th>Satisfied</th>
<th>Neutral</th>
<th>Dissatisfied</th>
<th>Very Dissatisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accurate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easy to find</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easy to understand</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Well organized</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applicable to your tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?  □ Yes  □ No

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

Name

Company or Organization

Phone No.